Fairness in Resource Allocation and Slowed-down Dependent Rounding *

David G. Harris† Thomas Pensyl‡ Aravind Srinivasan§ Khoa Trinh¶

Abstract

We consider an issue of much current concern: could fairness, an issue that is already difficult to guarantee, worsen when algorithms run much of our lives? We consider this in the context of resource-allocation problems; we show that algorithms can guarantee certain types of fairness in a verifiable way. Our conceptual contribution is a simple approach to fairness in this context, which only requires that all users trust some public lottery. Our technical contributions are in ways to address the k-center and knapsack-center problems that arise in this context: we develop a novel dependent-rounding technique that, via the new ingredients of "slowing down" and additional randomization, guarantees stronger correlation properties than known before.

*Research supported in part by NSF Awards CNS-1010789 and CCF-1422569, and by a research award from Adobe, Inc.

†Department of Computer Science, University of Maryland, College Park, MD 20742. Email: `davidgharris29@gmail.com`

‡Department of Computer Science, University of Maryland, College Park, MD 20742. Email: `tpensyl@cs.umd.edu`

§Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742. Email: `srin@cs.umd.edu`

¶Department of Computer Science, University of Maryland, College Park, MD 20742. Email: `khoa@cs.umd.edu`

Contents

1 Introduction

Fairness is a challenging issue to address. Starting with older work such as [21], there have been highly-publicized works on bias in allocating scarce resources – e.g., racial discrimination in hiring applicants who have very similar resumes [6]. Studies have considered this in realms including politics [7], and recent work notes (possible) bias in electronic marketplaces [4, 5]. The world of algorithms and machine learning brings new challenges (such as the distribution of the data used to train an ML classifier) here; see, e.g., [1] for several articles and resources. In this work, we show that algorithms can in fact guarantee fairness better than in the "offline" world, for basic clustering problems in resource allocation. Conceptually, we borrow from the notions of chance optimization and service-level agreements to propose a simple approach to guaranteed algorithmic fairness; on the technical side, we present new dependent-rounding techniques. Additionally, our models consider classical problems such as k-center from a new angle, and help study *flexible* resource-allocation that arises in cloud-based services.

1.1 Motivating examples: fair k-center and k-supplier

Consider the classical k-center problem. An instance $\mathcal{I} = (V, d, k)$ of this problem consists of a set V of n vertices, and a symmetric distance-metric d on V. Our goal is to choose a set $\mathcal{S} \subseteq V$ as "open centers" (or facilities). For any vertex i, we define $T_i = \min_{j \in \mathcal{S}} d(i, j)$. The goal is to choose $\mathcal{S}$ such that $|\mathcal{S}| \leq k$, to minimize the objective function $R := \max_{i \in V} T_i$. Note that there are only $\binom{n}{2}$ possible values for the optimal radius R. Thus, we can guess this value in $O(n^2)$ time. This NP-hard problem seems well-understood at first sight; we can find a solution with radius at most $2R$ in polynomial time, and no better is possible unless P $=$ NP; see, e.g., [16]. The variant where V is partitioned into clients and facilities, $\mathcal{S}$ must be some subset of the facilities, and where we only need service for the clients, is called k-supplier: its polynomial-time approximability is precisely 3 unless P $=$ NP [16].

An alternative perspective is that the client set V often corresponds to a set of autonomous entities; each $j \in V$ primarily is concerned with getting a center that is close enough. Unfairness in this context would be, say, a client that is always assigned a distance close to the worst-possible – $3R$ for k-supplier – even in multiple invocations of an algorithm for this problem. Indeed, similar problems arise in work that some of us are involved in, in health-care facility location for epidemic response. Especially in lower-GDP countries, this involves difficult tradeoffs about which sub-population will be far from their nearest facility. How can we guarantee, for instance, that no particular sub-population consistently receives bad service in this sense?

Theorems 5.9 and 4.3 give proposed starting points for guaranteeing fairness here. Motivated by the fact that different clients may have different tolerances to distance to their closest client, Section 5 considers the following problem: given a partition of V into clients and facilities where only clients need service, we aim for a randomized algorithm $\mathcal{A}$ that opens k facilities such that each client j has an open facility within distance r_j with probability at least p_j (or proves that such a distribution over k facilities does not exist). The r_j and p_j are additional input parameters here.

Unfortunately, we do not expect to develop such an algorithm $\mathcal{A}$: even for the special case where all the r_j are the same and all p_j are 1, the hardness of k-supplier shows that we cannot expect to even guarantee $(3 - \epsilon)r_j$ for all j, with probability p_j. Thus, we need to take an approximation-algorithms perspective to this fairness problem, as in [18, 19] for instance. Theorem 5.9 gives an algorithm $\mathcal{A}'$ that guarantees each user a facility within distance $3r_j$ with probability at least $0.8039p_j$ (assuming the demands are feasible).

As a variant of the probability of good service, one can also consider the expected quality of service, along with a guaranteed (probability-one) lower bound on the quality of service. Theorem 4.3 yields a randomized polynomial-time algorithm for k-center which guarantees, for each client $i \in V$, that $T_i \leq 3R$ with probability one, while $\mathbf{E}[T_i] \leq 1.592R$. Thus even in the short run — over $\Omega(\log n)$ independent invocations of the algorithm, when the problem needs to be solved multiple times as in the streaming application

below — all clients get much better average service than the optimal result (2-approximation) for k-center would indicate (in addition to a probability-one bound, such as the $3R$ here, on the worst quality of solution possible).

We see two ways of viewing such results, in addition to fairness. From a practical perspective, e-commerce applications such as *group buying* offer discounts on products, provided a minimum number of users sign up within a certain amount of time [2]. Such applications take the traditional optimization setup of satisfying a collection of users simultaneously and add the ingredient of autonomous users — those who are only concerned with the quality of service they require. Furthermore, cloud services such as streaming-on-demand require *flexible* facility location: such services can periodically (say, once every few days) shuffle their placement of videos in order to improve customer experience. Note the significant difference from traditional motivations for k-center such as placing fire stations, wherein the placement will likely persist for a long time. We can use Theorem 4.3 to ensure that, with high probability, each customer gets an average quality of service that is noticeably better than what is guaranteed by the standard optimal approximation ratio, while also ensuring that no individual shuffle leads to quite bad service for any user. Thus, a second way of viewing our results is as suggesting these viewpoints more generally in combinatorial-optimization research.

We note that, while running our algorithm $\mathcal{A}$ with random coins will guarantee good coverage probabilities and expectations, it is not necessary to trust algorithm $\mathcal{A}$ itself. In order to obtain a *verifiably* fair distribution, we can repeat $\mathcal{A}$ independently a suitable number $N = \Theta((1/\min_j p_j) \cdot (\log n)/\epsilon^2)$ of times, display this multi-set of N solutions to all the clients, and (perhaps using a publicly-trusted lottery) choose a solution uniformly at random from this multi-set. It is easy to see that with high probability, this multi-set will ensure similar fairness properties as the algorithm $\mathcal{A}$ itself. For instance, if $\mathcal{A}$ ensures that each client j has a probability p_j of being served within distance r_j, then the multi-set will, with high probability, ensure that each client j, at least an $(1-\epsilon)p_j$-fraction of solutions wherein j is served within distance r_j. Hence — as long as j trusts the lottery used for the uniformly random choice — j is convinced that its desired guarantee holds to within the ϵ relative error.

1.2 Dependent rounding

In dependent rounding, we aim to preserve certain marginal distributions, small martingale differences, and/or (negative) correlation properties while satisfying some constraints with probability one: the need for this naturally arises for us since we have some hard constraints such as placing at most k facilities, (almost) satisfying some m budget constraints, and ensuring $T_i \leq 3R$. A crucial tool for our algorithms will be a dependent-rounding algorithm from [22]. We quote this result here, which will be used in several places.

Proposition 1.1. *There exists an algorithm* DEPROUND*(y) which takes as input a vector $y \in [0,1]^n$, and in polynomial time outputs a set $Y \subseteq [n]$ with the following properties:*

(P1) $\Pr[i \in Y] = y_i$*, for all* $i \in [n]$*,*

(P2) $\lfloor \sum_{i=1}^n y_i \rfloor \leq |Y| \leq \lceil \sum_{i=1}^n y_i \rceil$ *with probability one;*

(P3) For any $S \subseteq [n]$*, we have* $\Pr[Y \cap S = \emptyset] \leq \prod_{i \in S}(1 - y_i)$*.*

When we run DEPROUND on a vector $(y_1, \ldots, y_n)$, we say that it *selects* i if $i \in Y$, and we say that Y are the *selected items*. We often write Y_i for the indicator variable of the event that $i \in Y$.

A final convention which will be useful to us is the following: suppose $(y_1, \ldots, y_n)$ is a vector and $S \subseteq [n]$, then we define $\text{DEPROUND}(y, S) \subseteq S$ to be $\text{DEPROUND}(x)$, where the vector x is formed by $x_i = y_i[i \in S]$. (We use the Iverson notation throughout this paper, so that for any Boolean predicate P we let $[P]$ be one if P is true and zero is P is false.)

Over the last two decades, increasingly-sophisticated dependent-rounding techniques have been used to obtain good solutions to optimization problems on matroid- and matroid-intersection- polytopes, subject in some cases to additional constraints; see, e.g., [3, 10, 13, 15, 22]. In the context of facility-location problems, Charikar and Li [11] applied dependent rounding in a clever way to obtain a 3.25-approximation for the classic k-median problem. More recently, Byrka et. al. [8] showed the "near-independence" property for small subsets of variables when running dependent rounding on a random permutation of the input vector. Dependent rounding is often used to round a fractional vector subject to (almost) satisfying a single linear constraint with non-negative coefficients [3, 22].

In a different context, Chekuri et. al. [12, 13] gave powerful rounding schemes which can round a point inside a matroid polytope (with the rounded coordinates finally being negatively correlated) or a point inside a matroid-intersection polytope (with good concentration bounds for any linear function of the rounded coordinates); however, their techniques are not applicable here as we also have to deal with clustering-type constraints arising naturally in facility-location problems.

In fact, one of our results (knapsack center: Theorem 3.1) requires a new dependent-rounding method which can preserve a set of *hard* "clustering contraints" and a set of *soft* knapsack constraints while guaranteeing near-negative-correlation among any subset of the variables. We guarantee this by our *Symmetric Randomized Dependent Rounding* (SRDR) technique, which utilizes two major modifications to [3, 22]: (i) updating the variables "symmetrically" and in a slowed-down manner with some additional randomization, and (ii) stopping the process "early" when there are only $O(1/\epsilon)$ fractional values left. (Note that in some applications, it suffices to obtain an "almost" integral solution. In our problem, the left-over fractional variables belong to disjoint clusters and all such fractional clusters can be rounded independently. Another notable example is the approach used in [8] for the k-median problem: the remaining fractional variables result in some $O(1)$ extra open facilities and a special "postprocessing" step by Li and Svensson [20] can be used to correct the solution.) We provide a direct application of SRDR in the context of the knapsack center problem, and believe that SRDR is of independent interest.

1.3 Our main technical contribution

We develop SRDR in Section 2. Given any "weight" vector $a \in \mathbb{R}^n$, a fractional vector $x \in [0,1]^n$, and a parameter t, the technique allows us to efficiently round x into an "almost" integral vector X – one with at most t fractional values left. As in standard dependent rounding [3, 22], the expected value of the X_i's and the weighted sum are preserved: $\mathbf{E}[X_i] = x_i$ for all $i \in [n]$ and $\sum a_i x_i = \sum a_i X_i$. Moreover, any subset of the variables has the following strong properties:

$$\mathbf{E}\left[\prod_{i\in S} X_i \prod_{j\in T}(1-X_j)\right] \leq \mathbf{E}\left[\Big(\prod_{i\in S} X_i \prod_{j\in T}(1-X_j)\Big)^p\right] \leq \left(\prod_{i\in S} x_i \prod_{j\in T}(1-x_j)\right)^p, \tag{1}$$

where $p = 1 - 1/(t+1)$, for any $S, T \subseteq [n]$ and $S \cap T = \emptyset$.

$$\mathbf{E}\left[\prod_{i\in S} X_i \prod_{j\in T}(1-X_j)\right] \geq \mathbf{E}\left[\Big(\prod_{i\in S} X_i \prod_{j\in T}(1-X_j)\Big)^p\right] \geq \left(\prod_{i\in S} x_i \prod_{j\in T}(1-x_j)\right)^p, \tag{2}$$

where $p = 1 + \max\left\{\frac{|S\cup T|-1}{t+1-|S\cup T|}, \frac{|S\cup T|-1}{t+1-|S\cup T|}\right\}$, if $|S \cup T| \leq t$.

In comparison, only the case where $S = \emptyset$ or $T = \emptyset$ is handled by [3, 22] for (1); systematic lower-bounds as in (2) are quite new.

Perhaps our most noteworthy technical contribution is from Section 3, where we consider fair allocation in the *knapsack center* problem. In the basic version of this problem, we have a cost M_i for opening each

$i \in V$, and aim for a solution of total cost at most some budget B that minimizes $\max_i T_i$; this problem has the best-possible approximation ratio being 3 [16, 17]. More recent work has considered this problem with m different cost functions, each with a budget; we aim for a solution that (almost) respects all of these budgets (i.e., almost satisfies m packing constraints) and minimizes $\max_i T_i$ [14].

Our main result is, for any fixed $\epsilon > 0$, a randomized polynomial-time algorithm with small budget violation that guarantees for each client i that $T_i \leq 3R$ with probability one, and $\mathbf{E}[T_i] \leq (1 + 2/e + \epsilon)R$. We do so by applying a type of dependent rounding in the context of a partition matroid $\mathcal{M}$. More formally, suppose we are given a fractional solution y which is in the matroid polytope of $\mathcal{M}$ and satisfies m additional packing constraints, we round y to a random Y such that: (i) Y is a base of $\mathcal{M}$, (ii) Y satisfies the packing constraints with at most $(1 + \epsilon)$ multiplicative violation, and (iii) for any set S with $\sum_{i \in S} y_i \geq 1$, we have $\Pr[\wedge_{i \in S}(Y_i = 0)] \leq 1/e + \epsilon$. This dependent rounding scheme has two major advantages over an independent rounding. First, it achieves a run-time exponent of roughly $1/\epsilon$, as compared to $1/\epsilon^2$ for independent rounding. Secondly, we achieve results for the important case $m = 1$ which satisfy the packing constraint exactly (this is discussed in Section 3.2). (The powerful framework of [13] cannot be applied as it could output an "almost-base" of $\mathcal{M}$.)

In Section 4, we study the fair k-center problem. An initial clustering of the clients based on the LP solution is a standard step in several facility-location algorithms. Section 4.3 takes the new step of forming "partial" clusters – those within which the total LP mass of facilities may be smaller than one. Our new idea is to "move" some opening mass of other facilities inside each cluster towards the center. Analysis of the partial-clustering algorithm along with detailed computer-assisted choice of parameters, leads to Theorem 4.3. Section 5 develops bicriteria approximation algorithms for k-supplier subject to the above-discussed "(r_j, p_j)" constraints.

1.4 Comparisons to related work

The near-independence property of dependent rounding was studied by Byrka et. al. [8]. Their idea is to randomly permute the vector $\mathbf{x}$ before applying the dependent rounding of [22]. Then they argue that any two variables are "far" from each other and unlikely to be rounded together in a single round, which implies that any small groups of the variables are nearly independent. Our results here are improvements in several ways. First, our upper-bound (1) only depends on the remaining number of fractional variables (i.e., is independent of the number of terms n and of the ratio $a_{\max}/a_{\min}$), and the "target" probabilities need not be bounded away from 0 or 1. Secondly, it does not require the weights to be non-negative as in [8]. Finally, we show in Section 3.5 that our technique can be generalized to work with multiple linear constraints while slightly violating these soft constraints and still guaranteeing the "near-negative-correlation" property.

Each iteration of the standard dependent-rounding technique of [22] co-rounds two variables x_1, x_2 such that the sum $a_1x_1 + a_2x_2$ is preserved and the expected values of x_1, x_2 do not change. If $a_1a_2 > 0$, then an increase in x_1 will lead to a decrease in x_2 for the sum to remain the same, and vice versa. This explains why we obtain negative correlation for all techniques in [8, 12, 15, 22]. If the weights are arbitrary, then it may be that $a_1a_2 < 0$ and so we may have positive correlation between the variables x_1, x_2.

Our SRDR technique employs the following ideas to reduce this positive correlation. First, we *randomly* pick a pair (x_1, x_2) to co-round in each iteration so that the probability that $a_1a_2 < 0$ is only about the $1/2$ in worst case. Next, instead of enforcing either x_1 or x_2 to be integral after a single step, we allow both x_1 and x_2 to remain fractional in some cases. For example, suppose $a_1 = +1, a_2 = -1, x_1 = 0.1$, and $x_2 = 0.2$. The normal approach will round (x_1, x_2) to $(0, 0.1)$ with probability $8/9$ and $(0.9, 1)$ with probability $1/9$. Our idea is to round this pair "symmetrically", getting $(0, 0.1)$ with probability $1/2$ and $(0.2, 0.3)$ with probability $1/2$; importantly, this symmetric amount of change (a parameter called δ in SRDR) is chosen globally, instead of on the particular a_i, x_i chosen (at random). All of these make our analyses of SRDR possible.

Dependent rounding has been used in the context of facility-location problems in [9, 11]. The result of [11] was a 3.25-approximation algorithm for the k-median problem. Their idea is to create clusters of mass at least $1/2$ and use a greedy matching approach to create pairs of bundles with mass at least 1. Next, they apply (standard) dependent rounding to open exactly one facility per cluster while preserving the cardinality constraint that at most k facilities can be opened. Using the negative correlation property, Charikar and Li show that the expected connection cost of any client j is at most 3.25 times its fractional connection cost.

In this work, we make substantial contributions toward the dependent rounding framework: our SRDR technique will work for any linear constraint, not just a cardinality constraint, and the bounds (1), (2) allow both S and T to be nonempty. The drawback is a small increase in the running time. While we may not have full negative correlation between the variables (indeed, this may be impossible for arbitrary weights a_i), our technique does guarantee the near-negative-correlation and near-independence properties. This may be good enough for many applications, especially if we can deal with some $O(1)$ left-over fractional variables as in the knapsack center problem.

1.5 Our models

Here we review our lottery models for the center-type problems. We note that all known approximation algorithms for the k-center and knapsack center in the literature are deterministic. For any such algorithm $\mathcal{A}$, it is not difficult to point out an instance in which the connection costs of almost all clients assigned by $\mathcal{A}$ actually match the worst-case bound. In Sections 3 and 4, we give the **fair knapsack-center algorithm** and the **fair k-center algorithm**, which not only achieve the usual worst-case bound by $\mathcal{A}$, but also guarantee a much better expected quality of service.

In the above model, the expected cost of client j is bounded in terms of R, which is the optimal *deterministic* radius for the corresponding center-type problem. It is also natural to ask the question "what is the optimal distribution $\mathcal{D}$ on the facilities to minimize the maximum **expected connection cost** over all clients?" We refer to this as the **center-lottery** model. For example, in the k-center-lottery problem, we aim to find a distribution $\mathcal{D}$ on the given facilities such that for any $\mathcal{S} \sim \mathcal{D}$, we have (a) $|\mathcal{S}| = k$ and that (b) $\max_j \mathbf{E}_{\mathcal{D}}[T_j]$ is minimized. The LP relaxation for this problem is as follows.

$$
\begin{aligned}
\text{minimize } & R \\
\text{subject to } & \sum_{i \in V} x_{ij} = 1 && , \forall j \in V \\
& x_{ij} \leq y_i && , \forall i \in V \\
& \sum_{i} d(i,j) x_{ij} \leq R, && \forall j \in V \\
& \sum_{i \in V} y_i = k, \\
& x_{i,j}, y_i \geq 0.
\end{aligned}
$$

This is indeed a relaxation, because for any optimal distribution $\mathcal{D}^*$, setting $y_i = \Pr_{\mathcal{D}^*}[i \text{ is chosen}]$ and $x_{ij} = \Pr_{D^*}[i \text{ is connected to } j]$ gives a feasible solution.

Let $C_j := \sum_i d(i,j) x_{ij}$ denote the fractional connection cost of j. Using the Charikar-Li algorithm [11], one can obtain a random set of k facilities such that $\mathbf{E}[T_j] \leq 3.25 C_j \leq 3.25 R$. In Section 3.4, we extend this to give a 3.25-approximation algorithm for the Multi Knapsack-center-lottery problem. We leave it as an open question whether one can improve the approximation ratio of 3.25 for this problem. (Interestingly, the "better" techniques in [8, 20] for the k-median problem, which are based on rounding a bi-point solution, do not seem to work here.)

2 Symmetric Randomized Dependent Rounding (SRDR)

In this section, we introduce the SRDR scheme and prove that it obeys certain correlation properties. Suppose we are given vector $x \in [0,1]^n$ and a weight vector $a \in \mathbb{R}^n$. Let $\mathrm{frac}(x) = \{i : 0 < x_i < 1\}$ denote the set of indices of fractional values of x. Also, for any set $I = \{i_1, i_2, \ldots, i_k\}$, let $x_I := \{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}$ denote the subvector of x indexed by I.

The algorithm to round x is as follows, where $\mathbf{e}_i$ is the unit vector in the i^{th} dimension in the standard basis.

Algorithm 1 $\textsc{Round}(x, a)$

1: $\delta \leftarrow \min_i \left\{|a_i| \min\{x_i, 1 - x_i\}\right\}$
2: Randomly choose a pair $i^*, j^* \in [n]$ where $i^* < j^*$
3: With probability $1/2$,
$$x \leftarrow x + (\delta/a_{i^*})\mathbf{e}_{i^*} - (\delta/a_{j^*})\mathbf{e}_{j^*},$$
else,
$$x \leftarrow x - (\delta/a_{i^*})\mathbf{e}_{i^*} + (\delta/a_{j^*})\mathbf{e}_{j^*}.$$
4: **return** x

Algorithm 2 $\mathrm{SRDR}(x, a, t)$

1: For each i such that $a_i = 0$, draw x_i as an independent Bernoulli-x_i variable.
2: **while** $|\mathrm{frac}(x)| > t$ **do**
3: $\quad x_{\mathrm{frac}(x)} \leftarrow \textsc{Round}(x_{\mathrm{frac}(x)}, a_{\mathrm{frac}(x)})$ # Apply $\textsc{Round}$ to fractional elements of x.
4: **return** x

Proposition 2.1. *Given vectors $x \in [0,1]^n$, $a \in \mathbb{R}^n$, and integer $t \geq 1$, the algorithm $SRDR(x, a, t)$ will return a vector $X \in [0,1]^n$ with at most t fractional values in expected $O(n^2)$ time. Moreover, the weighted sum and marginal probabilities are both preserved: $\sum_i a_i X_i = \sum_i a_i x_i$ with probability one, and $\mathbf{E}[X_i] = x_i$ for all $i \in [n]$.*

Proof. The runtime of SRDR is determined by the number of calls to $\textsc{Round}$. Consider a single such call. We know $\delta > 0$, since all elements of $x_{\mathrm{frac}(x)}$ are in $(0,1)$, and all elements of $a_{\mathrm{frac}(x)}$ are non zero (due to step 1 of SRDR). Let i' be a minimizer in step 1 of $\textsc{Round}$. If i' is chosen in step 2, then at least one of the two possible outcomes of step 3 will produce $x_{i'} \in \{0,1\}$. Thus, with probability at least $\frac{2}{\mathrm{frac}(x)} \cdot \frac{1}{2} \geq \frac{1}{n}$, each call to $\textsc{Round}$ will reduce the number of fractional elements of x by at least 1. In expectation, the process will terminate after $O(n(n-t)) = O(n^2)$ calls.

Step 1 of SRDR preserves both the marginal probability (by definition of a Bournoulli-p variable) and the weighted sum (because $a_i = 0$). Step 3 of $\textsc{Round}$ preserves both the weighted sum and the marginal probabilities of x_{i^*}, x_{j^*} by construction. By induction, these properties are preserved in the final output. $\square$

The main results of this section are summarized in the following Theorem 2.2:

Theorem 2.2. *Given vectors $x \in [0,1]^n$, $a = (a_1, \ldots, a_n) \in \mathbb{R}^n$, and $t \in \mathbb{N}$, there exists a randomized algorithm $\mathcal{A}$ which can round x in expected $O(n^2)$ time and return a vector $X \in [0,1]^n$ with at most t fractional values. Both the weighted sum and all the marginal probabilities are preserved: $\sum_i a_i X_i = \sum_i a_i x_i$ with probability one, and $\mathbf{E}[X_i] = x_i$ for all $i \in [n]$. Let S, T be disjoint subsets of $[n]$. Then:*

1. *We have the upper correlation bound:*

$$\mathbf{E}\left[\Big(\prod_{i\in S} X_i \prod_{j\in T}(1-X_j)\Big)^p\right] \le \left(\prod_{i\in S} x_i \prod_{j\in T}(1-x_j)\right)^p, \tag{3}$$

   where $p = 1 - 1/(t+1)$. Furthermore, if $a_i \ge 0$ and $a_j \le 0$ for all $i \in S$, $j \in T$, OR *if $a_i \le 0$ and $a_j \ge 0$ for all $i \in S$, $j \in T$, then the inequality holds for $p = 1$.*

2. *Let $S' = \{i \in S : a_i > 0\} \cup \{i \in T : a_i < 0\}$ and $T' = \{i \in S : a_i < 0\} \cup \{i \in T : a_i > 0\}$. If $|S'|, |T'| \le t$, then we have the lower correlation bound:*

$$\mathbf{E}\left[\Big(\prod_{i\in S} X_i \prod_{j\in T}(1-X_j)\Big)^p\right] \ge \left(\prod_{i\in S} x_i \prod_{j\in T}(1-x_j)\right)^p, \tag{4}$$

   where $p = 1 + \max\left\{\frac{|S'|-1}{t+1-|S'|}, \frac{|T'|-1}{t+1-|T'|}\right\}$. More weakly, if $|S \cup T| \le t$, it holds for $p = 1 + \frac{|S\cup T|-1}{t+1-|S\cup T|}$.

Many of the proofs, which are somewhat technical, are deferred to Appendix A.

Algorithmic application of SRDR. A typical application of dependent rounding is to convert a fractional solution, for example the solution to some LP relaxation, into an integral solution. For SRDR, this is typically a two-part process. First, the SRDR algorithm converts the fractional solution into a *nearly* integral solution, which preserves the weight function $\sum a_i x_i$, while approximately preserving the moments. Next, one must apply a problem-specific "end-game" to convert the partially integral solution to a fully-integral solution.

There are a number of possible strategies for this second step. One attractive option is to apply independent rounding; this can cause a small change in the weight $\sum a_i x_i$. However, since only the t fractional values of x_i are affected by this second rounding step, this will cause a much smaller change than if one applied independent rounding to the original fractional values.

Another possible option, which we will use in Section 3, is to force all fractional values of x to be zero or one in order to minimize $\sum_i a_i x_i$. This ensures that the final solution has smaller weight than the original fractional solution; again, since there are only t fractional values, which are the only entries of x affected by this discretization step, this process does not affect the large-scale behavior of the solution.

2.1 Upper bound on near-independence

Proposition 2.3. *Let $X = \textsc{Round}(x, a)$ for vectors $x \in (0,1)^n$ and $a \in (\mathbb{R} \setminus \{0\})^n$, $n \ge 2$. For any set $S \subseteq [n]$,*

$$\mathbf{E}\left[\Big(\prod_{i\in S} X_i\Big)^p\right] \le \Big(\prod_{i\in S} x_i\Big)^p$$

holds for $p = 1 - \frac{1}{n}$. Furthermore, if all weights in a_S have the same sign, the inequality holds for $p = 1$.

Proof. See Appendix A. □

Proposition 2.4. *Let $X = \textsc{Round}(x, a)$ for vectors $x \in (0,1)^n$ and $a \in (\mathbb{R} \setminus \{0\})^n$, $n \ge 2$. For any disjoint sets $S, T \subseteq [n]$,*

$$\mathbf{E}\left[\Big(\prod_{i\in S} X_i \prod_{j\in T}(1-X_j)\Big)^p\right] \le \Big(\prod_{i\in S} x_i \prod_{j\in T}(1-x_j)\Big)^p, \tag{5}$$

holds for $p = 1 - \frac{1}{n}$. *If* $a_i > 0$ *and* $a_j < 0$ *for all* $i \in S$, $j \in T$, OR *if* $a_i < 0$ *and* $a_j > 0$ *for all* $i \in S$, $j \in T$, *then 5 holds for* $p = 1$.

Proof. Define vectors x', a' and random vector X'' as

$$x'_i := \begin{cases} 1 - x_i & i \in T \\ x_i & i \notin T \end{cases}, \qquad a'_i := \begin{cases} -a_i & i \in T \\ a_i & i \notin T \end{cases}. \qquad X''_i := \begin{cases} 1 - X_i & i \in T \\ X_i & i \notin T \end{cases}. \tag{6}$$

Let $X' = \textsc{Round}(x', a')$. It is straightfoward to verify that X' and X'' have the same joint probability distribution. Then by Proposition 2.3,

$$\mathbf{E}\left[\Big(\prod_{i\in S} X_i \prod_{j\in T}(1 - X_j)\Big)^p\right] = \mathbf{E}\left[\Big(\prod_{i\in S\cup T} X''_i\Big)^p\right] \leq \Big(\prod_{i\in S\cup T} x'_i\Big)^p = \Big(\prod_{i\in S} x_i \prod_{j\in T}(1 - x_j)\Big)^p$$

for $p = 1 - \frac{1}{n}$. If $a_i > 0$ for all $i \in S$ and $a_j < 0$ for all $j \in T$, then all elements of $a'_{S\cup T}$ are positive. If $a_i < 0$ for all $i \in S$ and $a_j > 0$ for all $j \in T$, then all elements of $a'_{S\cup T}$ are negative. In either case, the above steps hold for $p = 1$. □

We can now prove the upper bound of Theorem 2.2:

Proof of Theorem 2.2(1). Define $S' := \{i \in S \mid a_i \neq 0\}$ and $T' := \{i \in T \mid a_i \neq 0\}$. Observe that for each $i \in (S \setminus S') \cup (T \setminus T')$, X_i is an independent Bournoulli-x_i, as determined by step 1 of SRDR. In this case, $\mathbf{E}[X_i^p] = \mathbf{E}[X_i] = x_i \leq x_i^p$ (for $p \leq 1$), and similarly $\mathbf{E}[(1 - X_i)^p] \leq (1 - x_i)^p$. Thus,

$$\begin{aligned} \mathbf{E}\Big[\Big(\prod_{i\in S} X_i \prod_{j\in T}(1 - X_j)\Big)^p\Big] &= \prod_{i\in S\setminus S'} \mathbf{E}\left[X_i^p\right] \prod_{i\in T\setminus T'} \mathbf{E}\left[(1 - X_i)^p\right] \mathbf{E}\Big[\Big(\prod_{i\in S'} X_i \prod_{j\in T'}(1 - X_j)\Big)^p\Big] \\ &\leq \prod_{i\in S\setminus S'} x_i^p \prod_{i\in T\setminus T'} (1 - x_i)^p \, \mathbf{E}\Big[\Big(\prod_{i\in S'} X_i \prod_{j\in T'}(1 - X_j)\Big)^p\Big]. \end{aligned} \tag{7}$$

Now, at each call to $\textsc{Round}$, $x_{\mathrm{frac}(x)}$ and $a_{\mathrm{frac}(x)}$ satisfy the conditions for Proposition 2.4, with $n = |\mathrm{frac}(x)| \geq t + 1 \geq 2$. Furthermore, by Jensen's inequality, (5) additionally holds for any $p \leq 1 - 1/|\mathrm{frac}(x)|$, namely for $p = 1 - 1/(t + 1)$. Then if $X = \mathrm{SRDR}(x, a, t)$, we can show by induction that

$$\mathbf{E}\Big[\big(\prod_{i\in S'} X_i \prod_{j\in T'}(1 - X_j)\big)^p\Big] \leq \big(\prod_{i\in S'} x_i \prod_{j\in T'}(1 - x_j)\big)^p. \tag{8}$$

Combining (7) and (8) yields (3) for $p = 1 - 1/(t + 1)$.

If $a_i > 0$ and $a_j < 0$ for all $i \in S'$, $j \in T'$, *OR* if $a_i < 0$ and $a_j > 0$ for all $i \in S'$, $j \in T'$, then (5) holds for $p = 1$, as do (8) and (3) by the same reasoning as above. Finally, we relax the strict inequalities as S and T may contain i with $a_i = 0$. □

2.2 Lower bound on near-independence

In this section, we show the lower bound in Theorem 2.

Proposition 2.5. *Let* $X = \textsc{Round}(x, a)$ *for vectors* $x \in (0, 1)^n$ *and* $a \in (\mathbb{R} \setminus \{0\})^n$, $n \geq 2$. *For set* $S \subseteq [n]$, *define* $S_1 := \{i \in S \mid a_i > 0\}$ *and* $S_2 := \{i \in S \mid a_i < 0\}$. *If* $|S_1|, |S_2| \leq n - 1$, *then*

$$\mathbf{E}\Big[\Big(\prod_{i\in S} X_i\Big)^p\Big] \geq \Big(\prod_{i\in S} x_i\Big)^p$$

holds for $p = 1 + \max\left\{\frac{|S_1|-1}{n-|S_1|}, \frac{|S_2|-1}{n-|S_2|}\right\}$.

Proof. See Appendix A. □

Proposition 2.6. *Let $X = \textsc{Round}(x, a)$ for vectors $x \in (0,1)^n$ and $a \in (\mathbb{R} \setminus \{0\})^n$, $n \geq 2$. Let $S, T \subseteq [n]$ be disjoint sets. Define $S' := \{i \in S \mid a_i > 0\} \cup \{i \in T \mid a_i < 0\}$ and $T' := \{i \in S \mid a_i < 0\} \cup \{i \in T \mid a_i > 0\}$. If $|S'|, |T'| \leq n - 1$, then*

$$\mathbf{E}\left[\Big(\prod_{i \in S} X_i \prod_{j \in T}(1 - X_j)\Big)^p\right] \geq \Big(\prod_{i \in S} x_i \prod_{j \in T}(1 - x_j)\Big)^p, \tag{9}$$

holds for $p = 1 + \max\left\{\frac{|S'|-1}{n-|S'|}, \frac{|T'|-1}{n-|T'|}\right\}$.

Proof. This proof is very similar to that of Proposition 2.4. Reusing the definitions for x',a', and X'' from (6), observe that $S' = \{i \in S \cup T \mid a'_i > 0\}$ and $T' = \{i \in S \cup T \mid a'_i < 0\}$. Then if $|S'|, |T'| \leq n - 1$, we may apply Proposition 2.5 to the show the following

$$\mathbf{E}\left[\Big(\prod_{i \in S} X_i \prod_{j \in T}(1 - X_j)\Big)^p\right] = \mathbf{E}\left[\Big(\prod_{i \in S \cup T} X''_i\Big)^p\right] \geq \Big(\prod_{i \in S \cup T} x'_i\Big)^p = \Big(\prod_{i \in S} x_i \prod_{j \in T}(1 - x_j)\Big)^p,$$

where $p = 1 + \max\left\{\frac{|S'|-1}{n-|S'|}, \frac{|T'|-1}{n-|T'|}\right\}$. □

Proof of Theorem 2.2(2). This proof is very similar to that of Theorem 2.2(1). Define $S'' := \{i \in S \mid a_i \neq 0\}$ and $T'' := \{i \in T \mid a_i \neq 0\}$. For each $i \in (S \setminus S'') \cup (T \setminus T'')$, X_i is an independent Bournoulli-x_i, so we have $\mathbf{E}[X_i^p] = \mathbf{E}[X_i] = x_i \geq x_i^p$ (for $p \geq 1$), and similarly $\mathbf{E}[(1 - X_i)^p] \geq (1 - x_i)^p$. Thus,

$$\mathbf{E}\left[\Big(\prod_{i \in S} X_i \prod_{j \in T}(1 - X_j)\Big)^p\right] \geq \prod_{i \in S \setminus S''} x_i^p \prod_{i \in T \setminus T''}(1 - x_i)^p \, \mathbf{E}\left[\Big(\prod_{i \in S''} X_i \prod_{j \in T''}(1 - X_j)\Big)^p\right]. \tag{10}$$

Now, consider a single call $\textsc{Round}(x_{\mathrm{frac}(x)}, a_{\mathrm{frac}(x)})$. Assuming $|S'|, |T'| \leq t$ as in the theorem statement, we have $|S' \cap \mathrm{frac}(x)| \leq |S'| \leq t \leq \mathrm{frac}(x) - 1$ and $|T' \cap \mathrm{frac}(x)| \leq |T'| \leq t \leq \mathrm{frac}(x) - 1$, so the conditions for Proposition 2.6 are met. Furthermore, by Jensen's inequality, (9) additionally holds for any $p \geq 1 + \max\left\{\frac{|S' \cap \mathrm{frac}(x)|-1}{\mathrm{frac}(x)-|S' \cap \mathrm{frac}(x)|}, \frac{|T' \cap \mathrm{frac}(x)|-1}{\mathrm{frac}(x)-|T' \cap \mathrm{frac}(x)|}\right\}$, namely for $p = 1 + \max\left\{\frac{|S'|-1}{t+1-|S'|}, \frac{|T'|-1}{t+1-|T'|}\right\}$. Then if $X = \mathrm{SRDR}(x, a, t)$, we can show by induction that

$$\mathbf{E}\left[\Big(\prod_{i \in S''} X_i \prod_{j \in T''}(1 - X_j)\Big)^p\right] \geq \Big(\prod_{i \in S''} x_i \prod_{j \in T''}(1 - x_j)\Big)^p. \tag{11}$$

Combining (10) and (11) yields (4).

Lastly, observe that $\max\{|S'|, |T'|\} \leq |S' \cup T'| = |S \cup T|$. This implies that $|S \cup T| \leq t$ is a stronger assumption than $|S'|, |T'| \leq t$, and also that $p = 1 + \frac{|S \cup T|-1}{t+1-|S \cup T|}$ is larger than $1 + \max\left\{\frac{|S'|-1}{t+1-|S'|}, \frac{|T'|-1}{t+1-|T'|}\right\}$ (and so is valid by Jensen's inequality). □

3 The (multi) knapsack center problem

In this section, we consider the (Multi) Knapsack Center problem, a generalization of k-center. An instance $\mathcal{I}$ of this problem consists of a set V of n vertices, a symmetric distance metric d on V, and an $m \times n$ weight matrix M, which corresponds to m non-negative weight functions (scaled so the corresponding budgets are

all equal to one). Our goal is to choose a set $\mathcal{S} \subseteq V$ of facilities ("centers") so that (1) all m knapsack constraints are satisfied — that is,

$$\sum_{i \in S} M_{ki} \leq 1$$

for all $k = 1, \dots, m$ and (2) the objective function $R := \max_{i \in V} T_i = \max_{i \in V} \min_{j \in \mathcal{S}} d(i,j)$ is minimized. The Knapsack Center problem (in the case $m = 1$) was first studied by Hochbaum & Shmoys in [16], under the name "weighted k-center". This gave a 3-approximation algorithm and proved that this is best possible unless P = NP; see also [17]. More recently, Chen et. al. [14] considered the case $m > 1$. They showed that this problem is not approximable to within any constant factor, and gave a pseudo 3-approximation algorithm which may violate all but one knapsack constraint by a factor of $(1 + \epsilon)$.

Let R denote the optimal radius. For the standard Knapsack Center problem with one constraint, we give a polynomial-time algorithm which returns a feasible solution such that (1) all vertices are within distance $3R$ from some chosen center and (2) almost all vertices have expected connection cost at most $(1 + 2/e)R \approx 1.74R$. For the Multi Knapsack Center problem, we show that it is possible to obtain a similar result while slightly violating the knapsack constraints. (Again, the violation is likely unavoidable because it is NP-hard to approximate this problem to within any constant factor.)

Theorem 3.1. *For any $\delta, \gamma > 0$, there exists an algorithm which runs in $n^{O\left(\frac{1}{\delta\gamma^2}\right)}$ time and returns a feasible solution the Knapsack Center problem (i.e., with $m = 1$ knapsack constraint) such that $T_i \leq 3R$ for all $i \in V$. Moreover, there is a set $U \subseteq V$ (which is deterministic, not a random variable), such that:*

1. *$|U| \geq (1 - \delta)n$,*
2. *$\forall i \in U, \ \ \mathbf{E}[T_i] \leq (1 + 2/e + \gamma)R$.*
3. *$\forall i \in V, \ \ T_i \leq 3R$ with probability one.*

Theorem 3.2. *For any $\gamma, \epsilon \in (0, 1/2)$, there is an algorithm for the Multi Knapsack Center problem (i.e., with $m \geq 1$ knapsack constraints) which runs in expected time $n^{O\left(\epsilon^{-1} m^2 \sqrt{\frac{\log(m/\gamma)}{\gamma}}\right)}$ and returns a solution $\mathcal{S}$ such that*

1. *$MY \leq (1 + \epsilon)\vec{1}$, where Y is the indicator vector of $\mathcal{S}$,*
2. *$\forall i \in V, \ \ \mathbf{E}[T_i] \leq (1 + 2/e + \gamma)R$.*
3. *$\forall i \in V, \ \ T_i \leq 3R$ with probability one.*

3.1 An LP relaxation

Suppose R is the optimal radius. We define the polytope $\mathcal{P}(\mathcal{I}, R)$ containing points (x, y) which satisfy the following constraints (C1)—(C7). Given a solution to this LP, our goal is to (randomly) round y to an integral solution.

(C1) $\sum_{i \in V: d(i,j) \leq R} x_{ij} = 1$ for all $j \in V$ (all clients should get connected to some center),

(C2) $x_{ij} \leq y_i$ for all $i, j \in V$ (vertex j can only connect to center i if it is open),

(C3) $My \leq \vec{1}$ (the m knapsack constraints),

(C4) $0 \leq x_{ij}, y_i \leq 1$ for all $i, j \in V$.

By splitting vertices we may ensure that we satisfy the additional constraints

(C5) For all $i, j \in V$, we have $x_{ij} \in \{0, y_i\}$,

(C6) For all $i \in V$, we have $x_{ii} = y_i$.

It will be useful to enforce an additional constraint on the maximum size of the entries of M:

Proposition 3.3. *For any $\rho > 0$, we can find a solution to the LP satisfying (C1) – (C6), and which satisfies the additional property*

(C7) For any $i \in V$, if $M_{ki} \geq \rho$ for any $k = 1, \dots, m$, we have $y_i \in \{0, 1\}$.

The running time for this process in $n^{O(m/\rho)}$.

Proof. We say that an item i is *big* if $M_{ki} \geq \rho$ for any $k = 1, \dots, m$. Suppose we fix any optimal solution $\mathcal{S}$. Observe that there can be at most m/ρ big facilities in S. We can guess the set of such facilities in time $n^{O(m/\rho)}$. For any big facility i that we guess is in S, we set $y_i = 1$; for any big facility that we guess is outside S, we set $y_i = 0$. This procedure will only check at most $n^{O(m/\rho)}$ possible cases. If our guess is correct, we have that $\mathcal{P}(\mathcal{I}, R) \neq \emptyset$. ☐

For any $j \in V$, let $F_j := \{i \in V : x_{ij} > 0\}$. We refer to these sets as *clusters*. It is easy to verify that $y(F_j) = 1$ due to (C5). We now form a subset $V' \subseteq V$ such that all the clusters $F_j, F_{j'}$ for $j, j' \in V'$ are pairwise disjoint, and such that V' is maximal with this property.

We also define

$$F_0 = \left\{ i \in V \mid y_i > 0, i \notin \bigcup_{j \in V'} F_j \right\}.$$

By property (C6), we have that $j \in F_j$ if $y_j > 0$, which implies that F_0 and V' are disjoint.

We now partition V into a set of *groups*. There are two types of groups. First, for each $j \in V'$, we define the group G_j to be simply F_j. Next, for each $j \in F_0$, we create a group G_j which consists of two items, namely j and a new dummy item Dummy(j), which has $y(\text{Dummy}(j)) = 1 - y_j$, which has distance ∞ from all $i \in V$, and which has cost zero according to all m knapsack constraints. (If we select this dummy item, it simply means that we do not choose to select item $j \in F_0$.) We let W denote the set of groups.

The following notation will be useful throughout: for any set $X \subseteq [n]$, we define frac(X) to be the set of items $i \in X$ such that $y_i \notin \{0, 1\}$; here y will always denote the *current* value of the vector $y \in [0, 1]^n$.

In the first step of both algorithms, we simplify y to reduce the number of fractional items in each G_j to at most $m + 1$ (this is automatically the case for $j \in F_0$). We do so using the following algorithm KnapsackIntragroupReduce:

Algorithm 3 KnapsackIntragroupReduce (y, X)

1: **while** $|\text{frac}(X)| > m + 1$ **do**
2: Let $\delta \in \mathbb{R}^n, \delta \neq 0$ be such that $M\delta = 0$, $\delta(X) = 0$, and $\delta_i = 0 \ \forall i \notin \text{frac}(X)$
3: Choose scaling factors $a, b > 0$ such that
- $y + a\delta \in [0, 1]^n$ and $y - b\delta \in [0, 1]^n$
- there is at least one entry of $y + a\delta$ which is equal to zero or one
- there is at least one entry of $y - b\delta$ which is equal to zero or one

4: With probability $\frac{b}{a+b}$, update $y \leftarrow y + a\delta$; else, update $y \leftarrow y - b\delta$.
5: **return** y

Proposition 3.4. *One can find a vector $\delta \in \mathbb{R}^n$ as claimed in line 2 of* KNAPSACKINTRAGROUPREDUCE.

Proof. This line is only executed when we have at least $m+2$ variables in X. On the other hand, we have only m knapsack constraints ($M\delta = 0$) and the additional linear constraint $\delta(X) = 0$. This system is underdetermined and the claim follows. $\square$

Proposition 3.5. *Suppose $y' =$ KNAPSACKINTRAGROUPREDUCE(y, X) for any $X \subseteq [n]$. Then $y'(X) = y(X)$ and $My = My'$ and $E[y'_i] = y_i$ for all $i \in V$.*

Proof. In each round of KNAPSACKINTRAGROUPREDUCE, we update y by either $y' := y + a\delta$ or $y' := y - b\delta$. Since δ is chosen so that $\delta(X) = 0$, we have $y'(X) = y(X) + a\delta(X) = y(X)$ or $y'(X) = y(X) - b\delta(X) = y(X)$. Similarly, as $M\delta = 0$ we have $My = My'$.

Also, for any $i \in V$ we have

$$\mathbf{E}[y'_i] = y_i + \frac{b}{a+b}(a\delta_i) - \frac{a}{a+b}(b\delta_i) = y_i.$$

The claim follows by induction on all iterations. $\square$

3.2 Proof of Theorem 3.1

Now we show that, when $m = 1$ (the standard Knapsack Center problem), one can satisfy the knapsack constraint with no violation while guaranteeing that the expected approximation ratio of at least $(1-\delta)n$ vertices is at most $1 + 2/e + \gamma$ for any $\gamma > 0$.

High-level ideas: First, by a preprocessing step (the PRUNE algorithm), we find a fractional solution in which each vertex i with $y_i > 0$ will only serve (fractionally) at most ϵn other vertices. (We do not really need to filter out "large" vertices having weight more than ρ in this case.) Next, we use the procedure KNAPSACKINTRAGROUPREDUCE to reduce the number of fractional variables inside G_j down to 2 for all $j \in W$. Note that the opening mass $y(G_j) = 1$ remains unchanged in the process. Define X_j to be the indicator for choosing the fractional vertex of higher weight in G_j. We then use SRDR to round vector $\mathbf{X}$ into an "almost" integral solution. That is, there will be at most some $O(1)$ groups G_j containing exactly two fractional vertices. Finally, we open the vertex with *smaller* weight in each fractional group.

We let M denote the weight function; since we are considering here the case that $m = 1$, we view M as an n-long vector where M_i is the weight of node $i \in V$.

(We note that it is formally possible for this process to return $\mathcal{S}$ which contains some dummy items. As these dummy items have infinite distance and zero weight, they contribute nothing and can simply be discarded.)

Proposition 3.6. *The procedure* PRUNE *runs in $n^{O(1/\epsilon)}$ time and will return EITHER an optimal solution OR a set $\mathcal{S}$ of open centers along with a fractional solution (x, y) for the residual instance $\mathcal{I}' = (V, d, m)$, in which each center i only serves $\leq \epsilon n$ other vertices fractionally (i.e., $|\{j \in V : x_{ij} > 0\}| \leq \epsilon n$ for all $i \in V$).*

Proof. Fix any optimal solution to the given instance. In Algorithm 4, whenever we find a vertex i which serves $\geq \epsilon n$ other vertices, we simply guess two cases: whether or not i is in the optimal solution. Note that for the guess "i is in the optimal solution", we open i and remove at least ϵn other vertices from the instance. (Because we assume that the budget constraint has RHS value of 1, this requires rescaling M to $\frac{M}{1-M_i}$.

In the other case, we remove i from V. Observe that if our guess is correct, $\mathcal{P}(\mathcal{I}', R)$ will not be empty in the next step. The algorithm stops when the current fractional solution satisfies all the properties in the claim.

Algorithm 4 PRUNE$(\epsilon, M, V, \mathcal{S}, R)$

1: **if** $V = \emptyset$ **then**
2:     **return** $\mathcal{S}$ *// we obtain an optimal solution in this case*
3: $\mathcal{I}' \leftarrow (M, V)$
4: **if** $\mathcal{P}(\mathcal{I}', R) \neq \emptyset$ **then**
5:     Compute a solution $(x, y) \in \mathcal{P}(\mathcal{I}', R)$
6:     **if** there exists some center i such that $M_i \leq 1$ and $|\{j \in V : x_{ij} > 0\}| \geq \epsilon n$ **then**
7:         Let $X \leftarrow \{j \in V : x_{ij} > 0\}$
8:         **return** PRUNE$(\epsilon, \frac{M}{1-M_i}, V \setminus X, \mathcal{S} \cup \{i\}, R)$ if it is not FALSE
9:         **return** PRUNE$(\epsilon, M, V \setminus \{i\}, \mathcal{S}, R))$
10:     **else**
11:         **return** (the solution (x, y), the set of remaining vertices V, the scaled weight matrix M, and the set of open centers $\mathcal{S}$)
12: **else**
13:     **return** FALSE

Algorithm 5 STANDARDKNAPSACKSRDR (M, V, t)

1: $(x, y, V, M, \mathcal{S}_0) \leftarrow$ PRUNE$(\epsilon, M, V, \emptyset, R)$ and **return** $\mathcal{S}$ if it already covers all nodes in V
2: **for each** $j \in V'$ **do** update $y \leftarrow$KNAPSACKINTRAGROUPREDUCE(y, G_j).
3: Let W' denote the set of groups $j \in W$ with $\text{frac}(G_j) = 2$.
4: **for each** $j \in W'$ **do**
5:     Suppose $G_j = \{i_1(j), i_2(j)\}$ and $M_{i_1(j)} \leq M_{i_2(j)}$
6:     Set $x_j \leftarrow y_{i_2(j)}$ and $a_j \leftarrow M_{i_2(j)} - M_{i_1(j)}$
7: $\mathbf{X} \leftarrow \text{SRDR}(\mathbf{x}, \mathbf{a}, t)$
8: **for each** $j \in W'$ **do**
9:     **if** $X_j = 1$ **then**
10:         Set $y_{i_1(j)} \leftarrow 0, y_{i_2(j)} \leftarrow 1$
11:     **else**
12:         Set $y_{i_1(j)} \leftarrow 1, y_{i_2(j)} \leftarrow 0$
13: **return** $\mathcal{S} = \mathcal{S}_0 \cup \{i \in V \mid y_i = 1\}$

To bound the running time of this algorithm, we can visualize its execution by a binary tree in which each node is either a vertex chosen in line 6 or a leaf. Each non-leaf node of the tree has two children corresponding to two decisions whether or not it is in the optimal solution. Indeed, the running time of the algorithm is bounded by the number of paths from the root to any leaf of this tree. Because (1) the length of any path is at most n and (2) the number of vertices chosen to be in the optimal solution in this path is at most $1/\epsilon$, the number of such paths is at most $n^{O(1/\epsilon)}$.

□

For the rest of this section, we will be working on the residual instance (M, V) returned by PRUNE. In the first part of the analysis, we show an upper bound on the probability that a given vertex $k \in V$ has no open facility in F_k. This is done by defining a natural potential function, which is an estimation for this probability. We will analyze its change after applying SRDR.

For each $j \in W$ we let $C_j = F_k \cap G_j$ be the set of vertices that vertex k is "interested in" from group j. Let y denote the current fractional solution at the beginning of line 7 of STANDARDKNAPSACKSRDR (just

before the execution of SRDR). We define a potential function:

$$S := \prod_{j \in W} \left(1 - y(C_j)\right).$$

Let x' be the value of the vector after executing SRDR. Let us next define a vector y' as follows: for each $j \in W$, we set $y'_{i1(j)} = x'_j, y'_{i2(j)} = 1 - x'_j$. For all other indices i we set $y'_i = y_i$. We define

$$S' := \prod_{j \in W} \left(1 - y'(C_j)\right).$$

Proposition 3.7. *Conditioned on any value of y and S, we have*

$$\mathbf{E}[(S')^{1-1/(t+1)}] \leq S^{1-1/(t+1)}.$$

Proof. Let $p = 1 - 1/(t+1)$. First, note that if $j \in W - W'$, then the execution of SRDR does not change the value of $y(C_j)$. Similarly, we have $y'(C_j) = y(C_j)$ if C_j has size zero, or if C_j contains both the items in G_j. So let us define the sets A, B as

$$A = \{j \in W : C_j = \{i_1(j)\}\}$$
$$B = \{j \in W : C_j = \{i_2(j)\}\}$$

Thus, it suffices to show that

$$\mathbf{E}[\prod_{j \in A \cup B} (1 - y'(C_j))^p] \leq \prod_{j \in A \cup B} (1 - y(C_j))^p$$

Note that $y(C_j) = x_j$ for $j \in B$ and $y(C_j) = 1 - x_j$ for $j \in A$, and similarly for y'. Thus, it suffices to show that

$$\mathbf{E}[\big(\prod_{j \in A} x'_j \prod_{j \in B} (1 - x'_j)\big)^p] \leq \big(\prod_{j \in A} x_j \prod_{j \in B} (1 - x_j)\big)^p$$

This is precisely the upper-correlation property of the SRDR (Theorem 2.2(1)). $\square$

Proposition 3.8. *We have that*

$$\mathbf{E}[S'] \leq (1/e)^{1-1/(t+1)}.$$

Proof. Let y^0 be the original fractional solution y at line 1 and $S_0 := \prod_{j \in W}(1 - y_0(C_j))$. We have $S_0 \leq \prod_{j \in W} e^{-\sum_{i \in C_j} y_i^0} = e^{-y^0(F_k)} = 1/e$. Next, since the clusters G_j are processed independently in line 1, and marginals are preserved by Proposition 3.5, we have

$$\mathbf{E}[S] = \prod_{j \in W} (1 - \mathbf{E}[y_i^1]) = \prod_{j \in W} (1 - y_i^0) = S_0 \leq 1/e.$$

Proposition 3.7 and Jensen's inequality give

$$\mathbf{E}[S'] \leq \mathbf{E}[(S')^{1-1/(t+1)}] \leq \mathbf{E}[S^{1-1/(t+1)}] \leq \mathbf{E}[S]^{1-1/(t+1)} \leq (1/e)^{1-1/(t+1)}.$$

$\square$

Proof of Theorem 3.1. **Feasibility of $\mathcal{S}$:** Recall that after calling the procedure PRUNE at line 1, the fractional solution (x, y) is feasible for the residual instance (V, M). It suffices to show that we do not violate the scaled knapsack constraint of the residual instance when rounding y. Proposition 3.5 ensures that the knapsack constraint is preserved (fractionally) by KNAPSACKINTRAGROUPREDUCE.

We can interpret lines 8 – 12 of STANDARDKNAPSACKSRDR as a two-part process. First, we set $y_{i_1(j)} = 1 - X_j$, $y_{i_2(j)} = X_j$. Next, for each $j \in W'$ with $X_j \in (0, 1)$, we set $y_{i_1(j)} = 1, y_{i_2(j)} = 0$. We refer to the latter step as *rounding down* j. Every aspect of the STANDARDKNAPSACKSRDR preserves the marginal weight $\sum_i M_i y_i$, except this rounding-down step. Since $M_{i_1(j)} \leq M_{i_2(j)}$ for every $j \in W'$, the rounding-down step can only decrease $\sum_i M_i y_i$. As a consequence of this, at the end of STANDARDKNAPSACKSRDR, we have $\sum_i M_i y_i \leq \sum_i M_i y_i^0$. Thus $\sum_{i \in \mathcal{S}} M_i \leq \sum_{i \in \mathcal{S}_0} M_i + \sum_i M_i y_i^0 \leq 1$.

Cost analysis: For any $k \in V$, we have $T_k \leq R$ if there is a facility opened in F_k, and $T_k \leq 3R$ otherwise. (By construction, $F_k \cap F_j \neq \emptyset$ for some $j \in V'$ and we always open one center inside G_j. Then the distance from k to this center is at most $d_{kj} + R \leq 3R$ by triangle inequality.) Note that there are at most $2t$ fractional vertices in V. For each vertex j, let q_j denote the probability that j is adjacent to such an unrounded vertex.

$$
\begin{aligned}
\sum_{j \in V} q_j &\leq \sum_{j \in V} \sum_{k \in F_j} \Pr[k \text{ is fractional}] && \text{(by the union-bound)} \\
&= \sum_{k \in V} \Pr[k \text{ is fractional}] \sum_{j \in F_k} 1 \\
&\leq (\epsilon n) \sum_{k \in V} \Pr[k \text{ is fractional}] && \text{(by the pre-processing step)} \\
&\leq (2t)\epsilon n
\end{aligned}
$$

We say that a vertex is good if $q_j \leq 1/t$ and bad otherwise. Then the number of bad vertices is at most $2t^2 \epsilon n$. We let U be the set of good vertices. Let y'' be the vector y after finishing the for-loop at lines 9–10 and let

$$
S'' := \prod_{j \in W} (1 - y''(C_j)).
$$

Now fix any $k \in U$. We have that $F_k \cap \mathcal{S} = \emptyset$ iff $S'' = 1$, and $S'' = 0$ otherwise. Let $\mathcal{E}$ denote the event that there are no fractional vertices in F_k before line 9. Note that $\Pr[\mathcal{E}] = 1 - q_k \geq 1 - 1/t$. Then, for t large enough,

$$
\begin{aligned}
\Pr[\mathcal{S} \cap F_k = \emptyset] &= \Pr[\mathcal{S} \cap F_k = \emptyset \wedge \mathcal{E}] + \Pr[\mathcal{S} \cap F_k = \emptyset \wedge \neg\mathcal{E}] \\
&\leq \Pr[S'' = 1 \wedge \mathcal{E}] + \Pr[\neg\mathcal{E}] \\
&\leq \Pr[S' = 1 \wedge \mathcal{E}] + 1/t && \text{(if } \mathcal{E} \text{ then } S' = S'') \\
&\leq \mathbf{E}[S'] + 1/t \\
&\leq (1/e)^{1 - 1/(t+1)} + 1/t \leq 1/e + 2/t
\end{aligned}
$$

Thus,

$$
\mathbf{E}[T_k] \leq R + (2R) \Pr[\mathcal{S} \cap F_k = \emptyset] \leq (1 + 2/e + 2/t)R
$$

Now for any $\delta, \gamma > 0$, by setting $t = 2/\gamma$ and $\epsilon = \frac{\delta}{2t^2}$, the number of bad vertices is $\leq \delta n$ and, for any $k \in U$ we have $\mathbf{E}[T_k] \leq (1 + 2/e + \gamma)R$. The running time is $n^{O(1/\epsilon)} = n^{O\left(\frac{1}{\delta\gamma^2}\right)}$.

□

3.3 Independent rounding for Multi Knapsack-center

We will now consider a simple algorithm for the multi knapsack center problem based on independent rounding. This method can be used to be a 3.25-approximation algorithm for multi-knapsack-center-lottery as well. Toward the end of this subsection, we discuss some of the drawbacks of this method.

Algorithm 6 INDEPENDENTROUND (y, V', M)

1: $\mathcal{S} \leftarrow \emptyset$
2: **for** $i \in F_0$ **do**
3: With probability y_i, $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$
4: **for** $j \in V'$ **do**
5: Randomly pick a vertex X_j in F_j s.t. $\Pr[X_j = i] = y_i$ for all $i \in F_j$ // *recall that* $y(F_j) = 1$
6: $\mathcal{S} \leftarrow \mathcal{S} \cup \{X_j\}$
7: **return** $\mathcal{S}$

For any vertex $i \in V$, let Y_i be the indicator variable for the event that $i \in \mathcal{S}$. By construction, we have $\mathbf{E}[Y_i] = y_i$ and all variables Y_i are negatively correlated.

Proposition 3.9. *For any $\epsilon \in (0, 1)$, with probability at least $1 - m \exp\left(-\frac{\epsilon^2}{3\rho}\right)$, we have that $\sum_{i \in \mathcal{S}} M_{ki} \leq 1 + \epsilon$ for all $1 \leq k \leq m$. (We write this more compactly as $MY \leq 1 + \epsilon$.)*

Proof. The k^{th} weight function $\sum_{i \in \mathcal{S}} M_{ki} = M_k Y$ is a sum of negatively-correlated variables, each of which is bounded in $[0, \rho]$ and which has mean ≤ 1. Hence as the Chernoff-Hoeffding bound holds under negative correlation, we have

$$\Pr[M_k Y \geq 1 + \epsilon] = \Pr\left[\frac{M_k Y}{\rho} \geq (1 + \epsilon) \times \frac{1}{\rho}\right] \leq e^{-\epsilon^2/(3\rho)}.$$

Taking a union bound over all m constraints, the total probability that any of them is violated by more than ϵ is at most $me^{-\epsilon^2/(3\rho)}$. □

Proposition 3.10. *For any vertex $i \in V$, we have the conditional expectation*

$$\mathbf{E}[T_i \mid MY \leq (1 + \epsilon)\vec{1}] \leq \left(1 + \frac{2/e}{1 - m\exp(-\epsilon^2/(3\rho))}\right) R.$$

Proof. Note that $y(F_i) = 1$. So by negative correlation, the probability that there are no open centers in F_i, i.e. $F_i \cap \mathcal{S} = \emptyset$, is

$$\mathbf{E}\left[\prod_{j \in F_i} (1 - Y_j)\right] \leq \prod_{j \in F_i} (1 - y_j) \leq e^{-y(F_i)} = 1/e.$$

Hence, the probability that there are no such open centers, *conditioned on $MY \leq (1 + \epsilon)\vec{1}$*, is at most

$$\frac{1}{e\left(1 - m\exp\left(-\epsilon^2/(3\rho)\right)\right)}.$$

Also observe that $T_i \leq 3R$ with probability one. Therefore,

$$\mathbf{E}[T_i] \leq R + \frac{1}{e\left(1 - m\exp\left(-\epsilon^2/(3\rho)\right)\right)} \times (2R) = \left(1 + \frac{2/e}{1 - m\exp(-\epsilon^2/(3\rho))}\right) R.$$

□

Proposition 3.11. *For any $0 < \gamma \leq 1/2$ and $0 < \epsilon < 1$, there is an algorithm running in expected time $n^{O(m \log(m/\gamma)\epsilon^{-2})}$, which outputs a solution $\mathcal{S}$ satisfying*

$$\begin{aligned} &MY \leq (1+\epsilon)\vec{1} \\ &\forall i \in V, \mathbf{E}[T_i] \leq (1 + 2/e + O(\gamma))R. \end{aligned}$$

Proof. Set $\rho = \epsilon^2/(3\log(m/\gamma))$ and apply Proposition 3.3 to achieve a fractional solution y satisfying (C1) – (C7). Now repeatedly run $\mathcal{S} = \textsc{IndependentRound}(y)$ until the resulting solution satisfies $MY \leq 1 + \epsilon$; the first time it does so, return $\mathcal{S}$.

Observe that the resulting distribution on the random variables $T_1, \dots, T_n$ output by this process is the same as the distribution of $T_i \mid MY \leq (1+\epsilon)\vec{1}$ if we run $\textsc{IndependentRound}(y)$ once. So by Proposition 3.10 we have

$$\mathbf{E}[T_i \mid MY \leq (1+\epsilon)\vec{1}] \leq \left(1 + \frac{2/e}{1 - m\exp(-\epsilon^2/(3\rho))}\right) R.$$

Then $m \exp\left(-\frac{\epsilon^2}{3\rho}\right) \leq \gamma$ and hence $\mathbf{E}[T_i] \leq \left(1 + \frac{2/e}{1-\gamma}\right) R \leq (1 + 2/e + O(\gamma))R$.

Also, the number of repetitions of this process is a geometric random variable, with success probability equal to the probability that $MY \leq 1 + \epsilon$; this probability is $1 - \gamma \geq 1/2$. So we only need an expected constant number of iterations to succeed.

The overall cost is thus $n^{O(m/\rho)} = n^{O(m \log(m/\gamma)\epsilon^{-2})}$. □

The main drawback with this approach is that the running-time exponent has a quadratic dependence on $1/\epsilon$. This is due to the fact that we may end up opening quite a few centers by independent rounding. To make sure that we only violate the budget constraints by at most ϵ w.h.p, we have to get rid of all the centers having weight $\Omega(\epsilon^2)$. Also, note that this type of independent rounding cannot be used for the case $m = 1$ (the standard Knapsack Center problem) to give a fully-feasible solution (not violating the budget constraint) such that the expected cost of $(1-\delta)n$ clients are bounded by a factor of $(1 + 2/e + \gamma)$ times the optimal radius, because the budget constraint is violated with constant probability.

In Section 3.5, we discuss a more sophisticated dependent-rounding approach which only opens at most a (small) constant number of centers that may cause a violation of the knapsack constraints. This reduces the dependence of ϵ in the exponent of the running time to near-linear.

3.4 Independent rounding for Multi Knapsack-center-lottery

We can consider a lottery variant of the multi knapsack-center: we want to find a distribution $\mathcal{D}$ on the random variables Y, such that $MY \leq 1$ with probability one. The objective function in this case is $\max_i \mathbf{E}_{\mathcal{D}}[T_i]$.

Recall that a vertex is big if its weight is at least ρ in some knapsack constraint. We can use the overall strategy of Section 3.3, in which we first guess the set of big items to open, then form a LP conditioned on the open vertices, and then do independent rounding of the resulting solution. However, we must be more careful because there is no single "optimal" choice for big items; we must consider the full distribution of their values.

Theorem 3.12. *For any $\gamma \in (0, 1/2]$ and any $\epsilon \in (0, 1)$, there is an algorithm running in expected time $n^{O(m \log m/\gamma)\epsilon^{-2}}$ which returns a lottery distribution $\mathcal{D}$ satisfying*

$$\begin{aligned} MY \leq (1+\epsilon)\vec{1} \qquad &\textit{with probability one} \\ \forall i \in V, \mathbf{E}[T_i] \leq (3.25 + O(\gamma))&\textit{OPT} \end{aligned}$$

Proof. Let us define $\mathcal{U}$ to be the collection of all sets of big items which does not exceed the budget constraints; this has size at most $n^{O(m/\rho)}$. Consider the following LP:

$$\begin{aligned}
&\text{minimize } R \\
&\text{subject to } \sum_{i \in V} x_{U,i,j} = q_U && \forall i,j \in V, U \in \mathcal{U} \\
&x_{U,i,j} \le y_{U,i} && \forall i \in V, U \in \mathcal{U} \\
&\sum_{U \in \mathcal{U}} \sum_{i} d(i,j) x_{U,i,j} \le R && \forall j \in V \\
&\sum_{i \in V} M_{k,i} y_{U,i} \le q_U && \forall U \in \mathcal{U}, k \in [m] \\
&x_{U,i,j}, y_{U,i}, q_U \ge 0 \\
&y_{U,i} = 1 && \forall U \in \mathcal{U}, i \in U \\
&y_{U,i} = 0 && \forall U \in \mathcal{U}, i \in V \setminus U, i \text{ is big} \\
&\sum_{U \in \mathcal{U}} q_U = 1
\end{aligned}$$

This LP has size $n^{O(m/\rho)}$, and so it can be solved in time $n^{O(m/\rho)}$. We claim that for any lottery $\mathcal{D}$, the optimal solution to this LP has value at most $\max_i \mathbf{E}_{\mathcal{D}}[T_i]$.

For the lottery $\mathcal{D}$, let $\mathcal{V}$ be the random variable which is the set of opened big items. For each $U \in \mathcal{U}$, set $q_U = \Pr[\mathcal{V} = U]$. For each i and $U \in \mathcal{U}$, set $y_{U,i} = \Pr[\mathcal{V} = U$ and i is opened$]$. For each i,j and $U \in \mathcal{U}$, set $x_{U,i,j}$ to be the probability that $\mathcal{V} = U$ and that j connects to i.

The constraint $\sum_{i \in V} M_{k,i} y_{U,i} \le q_U$ needs some further explanation. Observe that with the above fractional assignment $y_{U,i}/q_U$ is the probability that i is opened, conditioned on $\mathcal{V} = U$. Since $MY \le 1$ with probability one, we have $\mathbf{E}[MY \mid \mathcal{V} = U] \le 1$ and hence $\sum_i M_{k,i} y_{U,i}/q_U \le 1$.

We now describe how to round a fractional solution to this LP. We first draw the random variable $\mathcal{V}$, in which each $U \in \mathcal{U}$ is selected with probability q_U. (This is valid as the q_U are non-negative and sum to one.) Next, after $\mathcal{V} = U$ is fixed, we define

$$y'_i = \min\{y_{U,i}/q_U, 1\}, \quad x'_{i,j} = x_{U,i,j}/q_U,$$

for each $i,j \in V$.

Finally, we use the Charikar-Li algorithm [11] to perform the rounding for the probabilities $x'_{i,j}, y'_i$. (In fact, we may use a simpler version of their algorithm, because we do not have any constraint on the number of open facilities.)

Let us analyze the expected performance of this procedure. Once we have conditioned on $\mathcal{V} = U$, the vectors x', y' are a valid solution to the following LP:

$$\begin{aligned}
&\sum_{i \in V} x'_{i,j} = 1 && , \forall j \in V \\
&x'_{i,j} \le y'_i && , \forall i \in V \\
&\sum_{i \in V} M_{k,i} y'_i \le 1, \forall U \in \mathcal{U} \\
&x'_{i,j}, y'_i, \ge 0.
\end{aligned}$$

Hence, the Charikar-Li algorithm ensures that $\mathbf{E}[T_i \mid \mathcal{V} = U] \leq 3.25 \sum_j x'_{i,j} d(i,j) = 3.25 \sum_j x_{U,i,j}/q_U$. Integrating over $\mathcal{V}$, we have

$$\begin{aligned}
\mathbf{E}[T_i] &\leq \sum_{U \in \mathcal{U}} \Pr[\mathcal{V} = U] \times 3.25 \sum_j (x_{U,i,j}/q_U) d(i,j) \\
&= 3.25 \sum_{U \in \mathcal{U}} \sum_j x_{i,j} d(i,j) \leq 3.25R
\end{aligned}$$

Next, let us consider the probability of the event $MY \geq 1 + \epsilon$. Once we condition on $\mathcal{V} = U$, then the random variables Y_i are negatively correlated. (Note that they are not unconditionally negatively correlated). Also, each term $M_{k,i} Y_i$ is either deterministic or is in the range $[0, \rho]$. Finally, we have that

$$\mathbf{E}\left[\sum_i M_{k,i} Y_i \,\middle|\, \mathcal{V} = U\right] = \sum_i M_{k,i} y'_i \leq \sum_i M_{k,i} y_{U,i}/q_U \leq 1$$

Thus by the Chernoff-Hoeffding bound for negatively-correlated random variables, $\Pr[M_k Y \geq 1 + \epsilon \,\big|\, \mathcal{V} = U] \leq e^{-\epsilon^2/(3\rho)}$. Integrating over $\mathcal{V}$, we have $\Pr[M_k Y \geq 1 + \epsilon] \leq e^{-\epsilon^2/(3\rho)}$.

In particular, setting $\rho = \epsilon^2/(3 \log(m/\gamma))$, we see that probability of at least $1 - \delta$ of satisfying $MY \leq 1 + \epsilon$ and the expected value $\mathbf{E}[T_i \mid MY \leq 1 + \epsilon] \leq (3.25 + O(\gamma))R$. $\square$

3.5 Proof of Theorem 3.2

Here we describe a generalization of SRDR for rounding a fractional solution of the Multi Knapsack Center problem. Again, we will round y to an "almost" integral vector. During this process, whenever a fractional variable becomes integral (i.e., $y_i = 0$ or $y_i = 1$), then it stays at that values for the remainder of the algorithm. Furthermore, if $y_i = 1$, then i will be put into $\mathcal{S}$. Thus the fractional clusters $\mathrm{frac}(G_j)$ will continually shrink; so $\mathrm{frac}(G_j)$ always denotes the vertices in G_j which are fractional for the *current* value of y.

Finally, we use dependent rounding (independently on each group) to completely transform y to an integral vector, which will be our solution $\mathcal{S}$.

We summarize these three steps via the following algorithm KNAPSACKDEPENDENTROUND (which we abbreviate as KDR).

Algorithm 7 KDR (y, M, t)

1: **for each** $j \in V'$ **do** update $y \leftarrow$KNAPSACKINTRAGROUPREDUCE(y, G_j);
2: **while** $\sum_{j \in W}(|\text{frac}(G_j)| - 1) > t$ **do**
3: $\quad J := \emptyset$
4: $\quad$**for each** $j \in W$ **do**
5: $\quad\quad$With probability $p_j = 3m \times \frac{|\text{frac}(G_j)|-1}{\sum_{u \in W}(|\text{frac}(G_u)|-1)}$, update $J \leftarrow J \cup \{j\}$
6: $\quad\quad$Select two distinct vertices $i_{j1}, i_{j2} \in \text{frac}(G_j)$ uniformly at random
7: $\quad$**if** $|J| \geq m + 1$ **then**
8: $\quad\quad$Choose $\delta \in \mathbb{R}^n, \delta \neq 0$ such that
- $M\delta = 0$, $y + \delta \in [0,1]^n$, and $y - \delta \in [0,1]^n$
- There is at least one entry of $y + \delta$ or $y - \delta$ which is equal to zero or one
- For each $j \in J$, $\delta_{i_{j1}} = -\delta_{i_{j2}}$
- $\delta_i = 0$ if i is not a selected vertex from some G_j where $j \in J$

9: $\quad\quad$With probability $1/2$, update $y \leftarrow y + \delta$; else, update $y \leftarrow y - \delta$
10: For each $j \in W$ select a vertex $X_j \in G_j$, wherein we select $X_j = i$ with probability y_i
11: **return** $\mathcal{S} = \{X_j \mid j \in W\}$

We first show that these procedures are well-defined and terminate in expected polynomial time.

Proposition 3.13. *One can find a vector $\delta \in \mathbb{R}^n$ as claimed in line 8 of* KDR.

Proof. For each $j \in J$, let $x_{i_{j1}} = U_j$ and $x_{i_{j2}} = -U_j$ for some variable U_j to be determined. Set all other entries of x to be zero. Again, the system of $Mx = 0$ has $|J| \geq m + 1$ variables U_j and only m constraints, hence is underdetermine. So let x be any non-zero solution vector. Let $a \in \mathbf{R}$ be maximal such that $y + ax \in [0,1]^n$ and $y - ax \in [0,1]^n$. Then one may verify that $a < \infty$ and setting $\delta = ax$ achieves the claimed result. $\square$

Proposition 3.14. *Suppose $y' =$* KDR (y, M, t)*. Then $y'(G_j) = y(G_j)$ for all $j \in W$ and $E[y'_i] = y_i$ for all $i \in V$, and $My = My'$.*

Proof. In each round of Algorithm 7 that y is changed, we update it by either $y' := y + \delta$ or $y' := y - \delta$. Note that δ is chosen so that, for any $j \in W$, $\delta(G_j) = \delta_{i_{j1}} + \delta_{i_{j2}} = 0$. Thus, $y'(G_j) = y(G_j)$.

Similarly, δ is chosen so that $M\delta = 0$, so that $My = My'$.

Also, for any $i \in V$, we have

$$\mathbf{E}[y'_i] = y_i + (1/2)\delta_i - (1/2)\delta_i = y_i.$$

The claim follows by induction on all iterations. $\square$

Proposition 3.15. *The probability distribution in line 10 of* KDR *is valid, i.e. $y(G_j) = 1$.*

Proof. In the LP solution, we have $y(G_j) = 1$ by construction (C5) and by construction of G_j. After line 1, we have $y(G_j) = 1$ by Proposition 3.5. Before line 9, we have $y(G_j) = 1$ by Proposition 3.14. $\square$

Proposition 3.16. *When we run* KDR(y, M, t) *with $t \geq 3m^2$, then for all iterations of Algorithm 7 the probability vector p is valid, i.e. $p \in [0,1]^n$.*

Proof. We clearly have $p_j \geq 0$.

Line 5 of that algorithm is only executed as long as $\sum_{u \in W}(|\text{frac}(G_u)| - 1) > t$. Also, after applying KNAPSACKINTRAGROUPREDUCE to all clusters, we have that $|\text{frac}(G_j)| \leq m + 1$ for each $j \in W$. The

fractional clusters can only shrink during the execution of Algorithm 7, and so $|\mathrm{frac}(G_j)| \leq m+1$ for all iterations of Algorithm 7. Thus we have

$$p_j = 3m \times \frac{|\mathrm{frac}(G_j)| - 1}{\sum_{u \in W}(|\mathrm{frac}(G_u)| - 1)} \leq 3m \times \frac{m}{t} \leq 1$$

as desired. □

Proposition 3.17. *In any iteration of* KDR*, conditional on all prior state, there is a probability of at least* $1/10$ *that a variable becomes rounded. Thus,* KDR *terminates with probability one and has expected poly*(n) *running time.*

Proof. In any round with $|J| \geq m+1$, there is at least a $1/2$ probability of producing at least one quantized variable. Also, observe that $|J|$ is a sum of independent random variables with mean $\sum_{j \in W} p_j = 3m$, and hence by the Chernoff-Hoeffding bound the probability that it is $\leq m$ is at most $e^{-m(2/3)^2} \leq 0.65$. Thus, there is a probability of at least $(1-0.65) \times (1/2) \geq 1/10$ that there is a rounded variable.

This implies that the expected number of iterations of that algorithm is $O(n)$. It is clear that individual steps of the algorithm can be executed in polynomial time. □

Proposition 3.18. *Let* y *denote the fractional vector immediately before line 10 of* KDR*. Then* $My \leq \vec{1}$*.*

Proof. In the original fractional solution we have $My \leq \vec{1}$. By Propositions 3.5,3.14, My does not change through any of the intermediate steps of the algorithm. □

Next, we will analyze $\mathbf{E}[T_k]$ for a given vertex $k \in V$. To do so, we we show an upper bound on the probability that there is no open facility in F_k. Recall that, for each $j \in W$, $C_j = F_k \cap G_j$. Let y denote the current fractional solution at the beginning of some round q of KDR. For each $j \in W$ we define $c_j := \sum_{i \in C_j} y_i$ and a potential function:

$$S = \prod_{j \in W} (1 - c_j)$$

In Appendix B, we show that S does not increase too much in expectation over the execution of KDR).

Proposition 3.19. *Let* S *denote the value of the potential function at the beginning of the dependent rounding process (immediately after line 1 of* KDR*), and let* S' *denote the value of the potential function* ***at the end of all rounds*** *(i.e. at line 10 of* KDR*, after terminating* $\sum_{j \in W}(|\mathrm{frac}(G_j)| - 1) \leq t$*. Then*

$$\mathbf{E}[S'] \leq S + \frac{180m^2}{t}.$$

Proof. See Appendix B. □

Proposition 3.20. *For any vertex* $k \in V$*, the probability that* $\mathcal{S} \cap F_k = \emptyset$ *when* KDR *terminates is at most* $1/e + \frac{180m^2}{t}$.

Proof. In this proof, let y^0 be the original fractional vector y which is the input of Algorithm KDR. Let y^1 be the modified vector y after executing lines 1. Let y^2 be the modified vector immediately before line 10. Finally, let $y^3 = Y$, where Y_i is the indicator variable for $\mathcal{S}$.

Now fix any vertex $k \in V$. If $k \in V'$ then there is an open facility in F_k with probability 1. Suppose $k \notin V'$. We define

$$S_0 := \prod_{j \in W} (1 - \sum_{i \in C_j} y_i^0)$$
$$S_1 := \prod_{j \in W} (1 - \sum_{i \in C_j} y_i^1)$$
$$S_2 := \prod_{j \in W} (1 - \sum_{i \in C_j} y_i^2)$$
$$S_3 := \prod_{j \in W} (1 - \sum_{i \in C_j} y_i^3)$$

Observe that $X_j \in G_j$ and $X_j \in \mathcal{S}$. As the sets G_j are all pairwise disjoint, then $\mathcal{S} \cap G_j = \{X_j\}$. This implies that $|\mathcal{S} \cap C_j|$ is equal to zero or one, for each $j \in W$. Also, the entries of y^3 are all zero or one. Hence, we have that $S_3 = 1$ if $S \cap F_k = \emptyset$, and $S_3 = 0$ otherwise. So, it suffices to show that $\mathbf{E}[S_3] \leq 1/e + \frac{180m^2}{t}$.

Now, note that we have $S_0 \leq \prod_{j \in W} e^{-\sum_{i \in C_j} y_i^0} = e^{-y(F_k)} = 1/e$.

Next, since the clusters G_j are processed independently in lines 1, and marginals are preserved by Proposition 3.5, we have

$$\mathbf{E}[S_1] = \prod_{j \in W} (1 - \mathbf{E}[y_i^1]) = \prod_{j \in W} (1 - y_i^0) = S_0.$$

Next, by Proposition 3.19, we have that

$$\mathbf{E}[S_2] \leq \mathbf{E}[S_1] + \frac{180m^2}{t}.$$

Finally, the clusters G_j are processed independently in line 10, and again marginals are preserved by Property (P1), so $\mathbf{E}[S_3] = \mathbf{E}[S_2]$.

In total, we have that $\mathbf{E}[S_3] \leq 1/e + \frac{180m^2}{t}$. ☐

Next, we use concentration bounds to show that there is a good probability of not violating the knapsack constraints by too much.

Proposition 3.21. *With probability at least* $1 - m \exp\left(-\frac{2\epsilon^2}{t\rho^2}\right)$, *we have that* $MY \leq (1+\epsilon)\vec{1}$.

Proof. Immediately before line 10 of KDR, we have $\sum_{j \in W} (|\text{frac}(G_j)| - 1) \leq t$; also by Proposition 3.18, we have $My \leq 1$. This means that there are at most t groups G_j with $\text{frac}(G_j) > 0$ (that is, the value of y for the vertices in that cluster are not completely integral).

Now suppose we condition on the state before line 10. $M_k Y$ can be viewed as a sum of independent random variables (for each $j \in W$, the value of M_{k,X_k}). However, at most t of these variables are random — the remaining are deterministic (if $\text{frac}(G_j) = 0$, then there is exactly one entry $k \in G_j$ with $y_k = 1$, and $X_j = k$).

Thus we can view $M_k Y$ is a weighted sum of $\ell \leq t$ independent random variables, in which each variable is bounded in the range $[0, \rho]$. We may apply Hoeffding's bound: the probability that such a sum exceeds its mean by ϵ is at most $\exp(-\frac{2\epsilon^2}{t\rho^2})$.

This is the probability of violating a single knapsack constraint. By the union bound, the total probability of violating any constraint is at most $m \exp(-\frac{2\epsilon^2}{t\rho^2})$. ☐

Proof of Theorem 3.2. By rescaling, it suffices to show that $\mathbf{E}[T_i] \leq (1 + 2/e + O(\gamma))R$. We will choose parameters ρ, t and obtain a fractional solution which satisfies (C1) – (C7). We will independently repeat KDR until $MY \leq (1+\epsilon)\vec{1}$; thus it suffices to show that after a single application of KDR we have

1. $MY \leq (1+\epsilon)\vec{1}$ with constant probability
2. $\mathbf{E}[T_i \mid MY \leq (1+\epsilon)\vec{1}] \leq (1 + 2/e + O(\gamma))R$.

We have $T_i \leq R$ if there is a facility opened in F_i, and $T_i \leq 3R$ otherwise. So, by Proposition 3.20, the expected distance is at most $(1 + 2(1/e + 180m^2/t))R$. By Proposition 3.21, the probability of satisfying (approximately) the knapsack constraints is at least $1 - m\exp(-\frac{2\epsilon^2}{t\rho^2})$. Hence, the expected distance conditional on satisfying knapsack constraints is at most

$$\mathbf{E}[T_i \mid MY \leq (1+\epsilon)\vec{1}] \leq \frac{1 + 2/e + 320m^2/t}{1 - m\exp\left(-\frac{2\epsilon^2}{t\rho^2}\right)} R.$$

Now set $t = m^2/\gamma$ and set $\rho = \frac{\epsilon\sqrt{\gamma}}{m\sqrt{\ln(m/\gamma)}}$. Then

$$m\exp\left(-\frac{2\epsilon^2}{t\rho^2}\right) = \gamma^2/m \leq 1/4$$

Thus

$$\mathbf{E}[T_i \mid MY \leq (1+\epsilon)\vec{1}] \leq \frac{1 + 2/e + 320\gamma}{1 - \gamma^2/m} R \leq R\big(1 + 2/e + O(\gamma)\big)$$

The work is dominated by solving the LP to ensure (C1) — (C7); by Proposition 3.3, this takes time $n^{O(m/\rho)} = n^{O\left(\epsilon^{-1}m^2\sqrt{\frac{\log(m/\gamma)}{\gamma}}\right)}$. □

4 The k-center problem

In this section, we consider a fair variant of the k-center problem that was defined in the introduction. Suppose we are given an instance $\mathcal{I} = (V, d, k)$ of this problem, and suppose we have guessed the optimal radius R. Our goal is a randomized polynomial-time algorithm that opens at most k facilities with probability one and ensures that the distance T_i from any given client i to the nearest open facility, is minimized (in a probabilistic sense). Theorems 4.1 and 4.3 summarize our results; both of these ensure that $T_i \leq 3R$ with probability one, and focus on minimizing $\mathbf{E}[T_i]$ subject to this constraint. We note that it is possible to ensure that $T_i \leq 2R$ with probability one, but we do not see any algorithm which ensures that $T_i \leq (3 - \Omega(1))R$ and also gives a non-trivial guarantee $\mathbf{E}[T_i] < (\max T_i) - \Omega(1)$.

For any $i \in V$, we say that i *is open* if i is placed into the solution set $\mathcal{S}$.

4.1 An LP relaxation

Consider the polytope $\mathcal{P}(\mathcal{I}, R)$ containing points (x, y) with the following constraints:

(A1) $\sum_{i \in V: d(i,j) \leq R} x_{ij} = 1$ for all $j \in V$ (all vertices should get connected to some center);

(A2) $x_{ij} \leq y_i$ for all $i, j \in V$ (vertex j can only connect to center i if it is open);

(A3) $\sum_{i \in V} y_i \leq k$ (at most k centers can be opened); and

(A4) $0 \leq x_{ij}, y_i \leq 1$ for all $i, j \in V$.

Since R is the optimal radius, $\mathcal{P}(\mathcal{I}, R)$ is not empty. Our approach will be to find a fractional solution in $\mathcal{P}(\mathcal{I}, R)$ and then use a randomized algorithm to convert it into an integral solution.

By splitting vertices as needed, we can ensure that we have a fractional solution which satisfies the additional properties

(A5) For all $i, j \in V$, we have $x_{ij} \in \{0, y_i\}$,

(A6) For all $i \in V$, we have $x_{ii} = y_i$.

For any $j \in V$, let $F_j := \{i \in V : x_{ij} > 0\}$. We refer to these sets as *clusters*, and we refer to j as the *cluster center* of the cluster F_j. By (A5) and (A1), we have $y(F_j) = 1$ for all j; also, if $y_j \neq 0$, then $j \in F_j$.

4.2 A rounding algorithm that opens full clusters

We will give randomized rounding schemes based on forming clusters centered around certain vertices. As a warm-up exercise, we consider a scheme based on opening full clusters; we then describe a more complicated scheme based on partial clusters which achieves a slightly better approximation guarantee.

In the first scheme, we let $V' \subseteq V$ be a set of vertices which has the property that all F_j for $j \in V'$ are pairwise disjoint, and such that V' is maximal with this property. (This can be formed in a greedy way.) We define $F_0 = V \setminus \bigcup_{j \in V'} F_j$; it is the set of "unclustered" vertices.

Let $q \in [0, 1]$ be a parameter to be determined. Our algorithm is as follows:

Algorithm 8 ROUND1 $\left(y, F_0, \bigcup_{j \in V'} F_j, q\right)$

1: $\mathcal{S} \leftarrow \emptyset$
2: **for** $j \in V'$ **do**
3: Randomly pick a vertex $X_j \in F_j$ and assign $\mathcal{S} \leftarrow \mathcal{S} \cup \{X_j\}$ according to the following distribution

$$\forall i \in F_j : \Pr[X_j = i] = \begin{cases} q + (1-q)y_i & \text{if } i = j \\ (1-q)y_i & \text{if } i \neq j \end{cases}$$

// This is a valid probability distribution, as $\sum_{i \in F_j} y_i = y(F_j) = 1$
4: Let $I_0 \leftarrow$ DEPROUND(y, F_0)
5: $\mathcal{S} \leftarrow \mathcal{S} \cup I_0$
6: **return** $\mathcal{S}$

Throughout this section, we let Y_i be an indicator variable for the event that center i is open. Observe that if $i \in F_0$, then the dependent rounding process ensures that $\mathbf{E}[Y_i] = y_i$. Note that $\sum_{i \in F_0} y_i = \sum_{i \in V} y_i - \sum_{i \in \cup_{j \in V'} F_j} y_i$. Because the clusters F_j are disjoint for $j \in V'$, we have $\sum_{i \in \cup_{j \in V'} F_j} y_i = \sum_{j \in V'} y(F_j) = |V'|$. Thus, $\sum_{i \in F_0} y_i \leq k - |V'|$; The algorithm opens $|V'|$ facilities (one per full cluster) in the for loop and the dependent rounding opens $\leq \lceil k - |V'| \rceil$ facilities in F_0, for a total of at most k facilities.

We next show that this process has the property that $\mathbf{E}[T_i]$ is small for any vertex i.

Theorem 4.1. *For $q = 0.464587$, Algorithm 8 returns a solution $\mathcal{S}$ such that, for any $i \in V$, we have $T_i \leq 3R$ with probability one and $\mathbf{E}[T_i] \leq 1.60793R$.*

Proof. Let D denote the set of all $j \in V'$ such that $F_i \cap F_j \neq \emptyset$. By maximality of V', we must have $F_i \cap F_j \neq \emptyset$ for some $j \in V'$; thus $D \neq \emptyset$. For each $j \in D$, set $r_j = y(F_i \cap F_j)$ and set $r_0 = y(F_i \cap F_0)$. As F_0 and $F_j \mid j \in D$ are all pairwise disjoint, we have $r_0 + \sum_{j \in D} r_j = y(F_i) = 1$.

For each $j \in D$, our rounding step opens at least one center $v \in F_j$. As every center in F_i has distance at most R to i, and all centers in j has distance at most $2R$ from each other, it follows that $d(i, v) \leq 3R$.

Now suppose that we open either some center $v \in F_i \cap F_j$ for $j \in D$, or $v \in F_i \cap F_0$; such a center has distance $d(v, k) \leq R$, and thus in such cases $T_i \leq R$. Letting A denote the set of centers

$$A = (F_0 \cap F_i) \cup \bigcup_{j \in D} (F_j \cap F_i),$$

a necessary condition for $T_i \geq 2R$ is that no centers in A are open. Using negative correlation of dependent rounding,

$$\begin{aligned}
\Pr[T_i \geq 2R] \leq \Pr[\text{no centers in } A \text{ are open }] &= \mathbf{E}\left[\prod_{j \in D}\left(1 - \sum_{v \in F_i \cap F_j} Y_v\right) \prod_{v \in F_i \cap F_0} (1 - Y_v)\right] \\
&\leq \prod_{j \in D}\left(1 - \sum_{v \in F_i \cap F_j} (1-q) y_v\right) \prod_{v \in F_i \cap F_0} (1 - y_v) \\
&\leq \prod_{j \in D} (1 - (1-q) r_j) \prod_{v \in F_i \cap F_0} e^{-y_v} \\
&= \prod_{j \in D} (1 - (1-q) r_j) \times e^{-1 + \sum_{j \in D} r_j} \\
&= (1/e) \prod_{j \in D} e^{r_j} (1 - (1-q) r_j).
\end{aligned}$$

Similarly, if for some $j \in D$ we open center j itself, then $d(i, j) \leq 2R$ and hence $T_i \leq 2R$. Thus, let A' denote $A \cup D$; a necessary condition for $T_i \geq 3R$ is that we do not open any center in A'.

$$\begin{aligned}
\Pr[T_i \geq 3R] \leq \Pr[\text{no centers in } A' \text{ are open }] &= \mathbf{E}\left[\prod_{j \in D}\left(1 - Y_j - \sum_{v \in F_i \cap F_j} Y_v\right) \prod_{v \in F_i \cap F_0} (1 - Y_v)\right] \\
&\leq \prod_{j \in D}\left(1 - q - \sum_{v \in F_i \cap F_j} (1-q) y_v\right) \prod_{v \in F_i \cap F_0} (1 - y_v) \\
&\leq \prod_{j \in D} (1 - q - (1-q) r_j) \prod_{v \in F_i \cap F_0} e^{-y_v} \\
&= (1/e) \prod_{j \in D} e^{r_j} (1 - q - (1-q) r_j).
\end{aligned}$$

Putting these together gives:

$$\mathbf{E}[T_i] \leq R\big(1 + 1/e \prod_{j \in D} e^{r_j} (1 - (1-q) r_j) + (1/e \prod_{j \in D} e^{r_j} (1 - q - (1-q) r_j))\big) \tag{12}$$

Let $s = \sum_{j \in D} r_j$ and $t = |D|$. Then we have

$$\begin{aligned}
\mathbf{E}[T_i] &\leq R\big(1 + e^{s-1} \prod_{j \in D} (1 - (1-q) r_j) + e^{s-1} \prod_{j \in D} (1 - q - (1-q) r_j)\big) \\
&\leq R\big(1 + e^{s-1} (1 - (1-q) s/t)^t + e^{s-1} (1 - q - (1-q) s)^t\big) \qquad \text{(AM-GM inequality)}.
\end{aligned}$$

This is now a function of a single real parameter $s \in [0,1]$ (as well as an integer parameter t). Some simple analysis, which we omit here, shows that $\mathbf{E}[T_i]$ attains a maximum value of $1.60793R$. $\square$

4.3 Improved bounds via partial clusters

Forming partial clusters. We can improve on Theorem 4.1 through a more complicated rounding process which involves *partial* clusters. There is no randomness involved in this step; the clusters are selected in a greedy fashion, producing an ordering $\pi(1), \dots, \pi(n)$ as follows:

Algorithm 9 GREEDYFORMCLUSTERS (y)

```
for j ∈ V do: do
    Let i ∈ V − {π(1), ..., π(j − 1)} that maximizes y(F_i − F_π(1) − ··· − F_π(j−1))
    Set π(j) ← i.
```

For $j \in [n]$, we let $G_j = F_{\pi(j)} - F_{\pi(1)} - \dots - F_{\pi(j-1)}$; we refer to G_j as a *cluster* and $c_j = \pi(j)$ as the *cluster center* of the partial cluster G_j. We let $z_j = y(G_j)$. We say that G_j is a *full cluster* if $z_j = 1$ and a *partial cluster* otherwise.

Now suppose we have fixed the set of clusters. We use the following randomized rounding strategy to select the centers. We begin by choosing two real numbers $Q_{\mathrm{f}}, Q_{\mathrm{p}}$ (short for *full* and *partial*); these are drawn according to a joint probability distribution (not independently), which we discuss later. We then apply the following process:

Algorithm 10 ROUND2 $\left(y, z, \bigcup_{j=1}^{n} G_j, Q_{\mathrm{f}}, Q_{\mathrm{p}}\right)$

$\mathcal{S} \leftarrow \emptyset$

$Z \leftarrow \text{DEPROUND}(z)$

for $j \in Z$: **do**

Randomly pick a vertex $X_j \in G_j$ and assign $\mathcal{S} \leftarrow \mathcal{S} \cup \{X\}$ according to the following distribution

$$\forall i \in G_j : \Pr[X_j = i] = \begin{cases} q_j + (1-q_j) y_i / z_j & \text{if } i = c_j \\ (1-q_j) y_i / z_j & \text{if } i \neq c_j \end{cases}$$

and where we define q_j as

$$q_j = \begin{cases} Q_{\mathrm{f}} & \text{if } z_j = 1 \\ Q_{\mathrm{p}} & \text{if } z_j < 1 \end{cases}$$

return $\mathcal{S}$

We will defer the technical analysis of this scheme to Appendix C, but we give here some intuitive motivation. Consider some vertex i. It may be beneficial to open the center of some cluster near i (if we do not open any facility in the ball of radius R around i, then doing so ensures $T_i = 2R$ instead of $T_i = 3R$). However, there is no benefit to opening the centers of multiple clusters. So, we would like to ensure that there is a significant negative correlation between opening the centers of distinct clusters near i.

Unfortunately, there does not seem to be any way to achieve this with respect to full clusters — as all full clusters "look alike," we cannot create a probability distribution with any significant negative correlation among the indicator random variables for opening their centers.

By taking advantage of partial clusters, we are able to break this symmetry. Every vertex i will have at least one full cluster in its neighborhood, and possibly some partial clusters as well. We will create a

probability distribution so that either partial clusters open their centers, or full clusters open their centers — while we allow both to occur simultaneously, there is a negative correlation between these two possibilities. This ensures that a vertex i is less likely to see multiple neighboring clusters open their centers, which in turn leads to an improved value of $\mathbf{E}[T_i]$.

Proposition 4.2. *The resulting set $\mathcal{S}$ of open facilities satisfies $|\mathcal{S}| \leq k$.*

Proof. Observe that, for each $j \in [n]$, we have $|\mathcal{S} \cap G_j| \leq 1$. Also, $|Z| \leq \lceil z(V) \rceil$ by the dependent rounding, and $z(V) = \sum_j y(F_{\pi(j)} - F_{\pi(1)} - \cdots - F_{\pi(j-1)}) = \sum_j y_j \leq k$. □

Theorem 4.3. *Suppose we set our parameters for the distribution on Q_f, Q_p as follows:*

$$(Q_f, Q_p) = \begin{cases} (0.4525, 0) & \textit{with probability } p = 0.773436 \\ (0.0480, 0.3950) & \textit{with probability } 1-p \end{cases}.$$

Then, $T_i \leq 3R$ with probability one, and $\mathbf{E}[T_i] \leq 1.592R$.

Proof. See Appendix C. □

5 The chance metric k-coverage problem

We define the Chance Metric k-Coverage (CMkC) problem as follows. Let $\mathcal{I} = (k, \mathcal{C}, \mathcal{F}, d, p, r)$ be an instance of CMkC. $\mathcal{C}$ is a set of clients, $\mathcal{F}$ is a set of facilities, and d is a distance metric over $\mathcal{C} \cup \mathcal{F}$. p and r are vectors containing target probability p_j and target radius r_j for each client $j \in \mathcal{C}$. Finally, k is the target (integer) number of open facilities. The CMkC problem asks for a probability distribution over sets of k open facilities such that, for each client j, there is at least one open facility within distance r_j with probability at least p_j.

In the special case that all $p_j = 1$ and r_j are uniform, CMkC is equivalent to the decision version of the k-supplier problem. Thus CMkC is NP-hard, so we consider approximations. For $\alpha \geq 1$ and $\beta \leq 1$, define an (α, β)-approximation algorithm to be one which either proves there is no such distribution, or returns a distribution over sets of k open facilities such that for each client j, there is an open facility within distance αr_j with probability at least βp_j. We call α the distance guarantee and β the chance guarantee. Since k-supplier is hard to $(3-\epsilon)$-approximate [16], CMkC is hard to $(3-\epsilon, 1)$-approximate. We may also see by reduction to Max Set Cover (which is hard to $(1 - \frac{1}{e} - \epsilon)$-approximate) that CMkC is hard to $(1, 1 - \frac{1}{e} - \epsilon)$-approximate.

For a vector y and index set S, we use the shorthand $y(S) := \sum_{i \in S} y_i$.

5.1 Preliminaries

For each $j \in \mathcal{C}$, define $B_j := \{i \in \mathcal{F} \mid d(i,j) \leq r_j\}$ to be the ball of facilities within j's target radius. Consider the polytope $\mathcal{P}(\mathcal{I})$ containing points $y = (y_1, \dots, y_n)$ with the following constraints.

(A1) $y(B_j) \geq p_j$ for all $j \in \mathcal{C}$,

(A2) $y(\mathcal{F}) = k$,

(A3) $0 \leq y_i \leq 1$ for all $i \in \mathcal{F}$.

Proposition 5.1. *If there exists a distribution $\mathcal{D}$ which is a solution to $\mathcal{I}$, then $\mathcal{P}(\mathcal{I})$ is nonempty.*

Proof. For each $i \in \mathcal{F}$, set $y_i = \Pr_{\mathcal{D}}[i \text{ is open}]$. Each client j must have at least one open facility in B_j with probability at least p_j. By the union bound, this probability is at most $y(B_j)$, so (A1) is satisfied. The expected number of open facilities is $\sum_{i \in \mathcal{F}} y_i$ and is at most k, so (A2) is satisfied. (A3) is clearly satisfied. We have demonstrated a point in $\mathcal{P}(\mathcal{I})$. $\square$

If we find the polytope to be empty, then we know there is no solution to $\mathcal{I}$. Otherwise, we can find a point y in $\mathcal{P}(\mathcal{I})$ in polynomial time. For the remainder of Section 2, we assume we have such a vector y and focus on how to round it to obtain an integral solution.

Finally, for each client $j \in \mathcal{C}$ define F_j to be some subset of B_j such that $y(F_j) = p_j$. (We may enforce that such a subset exists using a standard splitting step, in which we replace some facility i with two smaller facilities i_1, i_2 such that $y_i = y_{i_1} + y_{i_2}$.) Although not necessary, using F_j instead of B_j will simplify the algorithm descriptions and analyses.

Theorem 5.2. *There is a $(1 - \frac{1}{e}, 1)$-approximation algorithm for CMkC.*

Proof. Set $\mathcal{S} = \textsc{DepRound}(y)$. This satisfies $|\mathcal{S}| \leq \lceil \sum_{i=1}^n y_i \rceil \leq \lceil k \rceil = k$ as desired. Now for each $j \in \mathcal{C}$, the probability that all facilities in F_j are closed is

$$\Pr[\mathcal{S} \cap F_j = \emptyset] \leq \prod_{i \in F_j} (1 - y_i) \leq \prod_{i \in F_j} e^{-y_i} = e^{-y(F_j)} = e^{-p_j}.$$

Therefore, with probability at least $1 - e^{-p_j}$, there will be a facility open in F_j (which is within distance r_j). Then j has local distance guarantee 1, and local chance guarantee $\frac{1-e^{-p_j}}{p_j}$. Then the global chance guarantee is $\min_{0 \leq p_j \leq 1} \frac{1-e^{-p_j}}{p_j} = 1 - \frac{1}{e}$. $\square$

5.2 Bundling

As previously mentioned, it is NP-hard to obtain an $(1, 1 - \frac{1}{e} + \epsilon)$-approximation. Thus, to improve the chance guarantee, we must relax the distance guarantee. To achieve this we first observe the previous algorithm is tight only when $p_j = 1$; for small p_j, the chance factor is much better. Thus we ignore clients with small p_j. Among remaining clients, we choose a sparse set to form *bundles* around, concentrating negative correlation within each bundle, and giving a stronger guarantee than (P3).

Algorithm 11 $\textsc{Filter}(z)$

1: $\mathcal{C}_z \leftarrow \{j \in \mathcal{C} \mid p_j \geq z\}$
2: $\mathcal{C}' \leftarrow \emptyset$
3: **while** $\mathcal{C}_z \neq \emptyset$ **do**
4: $\quad j' \leftarrow \arg\min_{j \in \mathcal{C}_z} r_j$
5: $\quad \mathcal{C}' \leftarrow \mathcal{C}' \cup \{j'\}$
6: $\quad \mathcal{C}_z \leftarrow \mathcal{C}_z \setminus \{j \in \mathcal{C} \mid F_j \cap F_{j'} \neq \emptyset\}$
7: **return** $\mathcal{C}'$

We define $F_0 = \mathcal{F} - \bigcup_{j \in \mathcal{C}'} F_j$. These are the *unclustered* vertices.

Proposition 5.3. $\textsc{Filter}$*(z) produces a set $\mathcal{C}' \subseteq \mathcal{C}$ such that:*

(Q1) $p_j \geq z$ *for all* $j \in \mathcal{C}'$*;*

(Q2) $\{F_j\}_{j \in \mathcal{C}'}$ *is pairwise disjoint;*

Algorithm 12 ROUNDBUNDLES(y, z)

1: For each $j \in \mathcal{C}$, select some $X_j \in F_j$, where the probability distribution on X_j is given by

$$\Pr(X_j = i) = y_i / p_j$$

(This is valid probability distribution as $y(F_j) = p_j$.)

2: $\mathcal{C}' \leftarrow$ FILTER(z)

3: Define the vector $q \in [0,1]^{\mathcal{F}}$ by

$$q(j) = \begin{cases} y_j & \text{if } j \in F_0 = \mathcal{F} - \bigcup_{k \in \mathcal{C}'} F_i \\ p_j & \text{if } j \in F_i \text{ and } j = X_i \text{ for } i \in \mathcal{C}' \\ 0 & \text{if } j \in F_i \text{ and } j \neq X_i \text{ for } i \in \mathcal{C}' \end{cases}$$

4: Return $\mathcal{S} =$ DEPROUND(q)

(Q3) There exists a function $\sigma(j) : \{j \in \mathcal{C} \mid p_j \geq z\} \to \mathcal{C}'$ such that for all valid j and $i \in F_{\sigma(j)}$, $d(i,j) \leq 3r_j$.

Proof. (Q1) follows directly from line 1, and (Q2) follows from line 6. To show (Q3), define $\sigma(j)$ to be the value of j' when j was removed from $\mathcal{C}_z$ in line 6 (note $\sigma(j) = j$ for $j \in \mathcal{C}'$). Then by line 4, $r_{j'} \leq r_j$. Also, $\exists i' \in F_j \cap F_{\sigma(j)}$. So for any $i \in F_{\sigma(j)}$, we have (by triangle inequality) $d(i,j) \leq d(i,\sigma(j)) + d(\sigma(j), i') + d(i', j) \leq 2r_{\sigma(j)} + r_j \leq 3r_j$. □

Proposition 5.4. *The algorithm* RoundBundles *opens at most k facilities.*

Proof. By (P2), we have

$$|\mathcal{S}| \leq \lceil \sum_{j \in \mathcal{F}} q(j) \rceil = \lceil \sum_{j \in F_0} y_j + \sum_{i \in \mathcal{C}'} \sum_{j \in F_i} q(i) \rceil = \lceil y(F_0) + \sum_{i \in \mathcal{C}'} y(F_i) \rceil = \lceil y(\mathcal{C}) \rceil \leq k$$

□

Proposition 5.5. *For any $S \subseteq \mathcal{F}$, we have*

$$\Pr(\mathcal{S} \cap S = \emptyset) \leq \prod_{i \in S \cap F_0} (1 - y_i) \prod_{j \in \mathcal{C}'} (1 - y(S \cap F_j)).$$

Proof. For each $i \in \mathcal{F}$ we let Y_i be the indicator variable for the event that $i \in \mathcal{S}$. Note that

$$[\mathcal{S} \cap S = \emptyset] = \prod_{i \in S \cap F_0} (1 - Y_i) \prod_{j \in \mathcal{C}' : X_j \in S} (1 - [j \in J])$$

Now, if we condition on the variables X_j, we have that

$$\begin{aligned} \Pr(\mathcal{S} \cap S = \emptyset \mid X_j) &= \mathbf{E}[\prod_{i \in S \cap F_0} (1 - Y_i) \prod_{j \in \mathcal{C}' : X_j \in S} (1 - [j \in J])] \\ &\leq \prod_{i \in S \cap F_0} \mathbf{E}[1 - Y_i] \prod_{j \in \mathcal{C}' : X_j \in S} (1 - \mathbf{E}[j \in J]) \qquad \text{by (P3)} \\ &= \prod_{i \in S \cap F_0} (1 - y_i) \prod_{j \in \mathcal{C}' : X_j \in S} (1 - p_j) \qquad \text{by (P1)} \end{aligned}$$

Thus integrating over X_j gives:

$$\begin{aligned}\Pr(\mathcal{S} \cap S = \emptyset \mid X_j) &\le \mathbf{E}[\prod_{i \in S \cap F_0} (1 - y_i) \prod_{j \in \mathcal{C}' : X_j \in S} (1 - p_j)] \\ &= \mathbf{E}[\prod_{i \in S \cap F_0} (1 - y_i) \prod_{j \in \mathcal{C}'} (1 - [X_j \in S] p_j)] \\ &= \prod_{i \in S \cap F_0} (1 - y_i) \prod_{j \in \mathcal{C}'} (1 - y(S \cap F_j)/p_j \times p_j)] \qquad \text{as each } X_j \text{ is independent} \\ &= \prod_{i \in S \cap F_0} (1 - y_i) \prod_{j \in \mathcal{C}'} (1 - y(S \cap F_j))\end{aligned}$$

□

Proposition 5.6. *For any $j \in \mathcal{C}$ and $j' \in \mathcal{C}'$ we have:*

(R1) $\Pr[\mathcal{S} \cap F_j = \emptyset] \le e^{-p_j}$,

(R2) $\Pr[\mathcal{S} \cap (F_j \cup F_{j'}) = \emptyset] \le e^{-(p_j - p_{j'})}(1 - p_{j'})$.

Proof. Apply Proposition 5.5 with $S = F_j$ to obtain:

$$\begin{aligned}\Pr(\mathcal{S} \cap F_j = \emptyset) &\le \prod_{i \in F_j \cap F_0} (1 - y_i) \prod_{i \in \mathcal{C}'} (1 - y(F_j \cap F_i)) \le \prod_{i \in F_j \cap F_0} e^{-y_i} \prod_{i \in \mathcal{C}'} (e^{-y(F_j \cap F_i)}) \\ &= e^{-y(F_j)} \qquad \text{as } F_0 \sqcup \bigsqcup_{i \in \mathcal{C}'} F_i = \mathcal{F} \\ &\le e^{-p_j}\end{aligned}$$

thus proving (R1).

Apply Proposition 5.5 with $S = F_j \cup F_{j'}$ to obtain:

$$\begin{aligned}\Pr(\mathcal{S} \cap (F_j \cup F_{j'}) = \emptyset) &\le \prod_{i \in (F_j \cup F_{j'}) \cap F_0} (1 - y_i) \prod_{i \in \mathcal{C}'} (1 - y((F_j \cup F_{j'}) \cap F_i)) \\ &\le \prod_{i \in (F_j \cup F_{j'}) \cap F_0} e^{-y_i} \times (1 - y((F_j \cup F_{j'}) \cap F_{j'})) \times \prod_{i \in \mathcal{C}', i \neq j'} (e^{-y((F_j \cup F_j') \cap F_i)}) \\ &= e^{-y(F_j \cup F_{j'})} \times e^{y(F_{j'})}(1 - y(F_{j'})) = e^{-y(F_j - F_{j'})}(1 - y(F_{j'}))\end{aligned}$$

Now note that $y(F_j) = p_j$ and $y(F_{j'}) = p_{j'}$ so that $e^{-y(F_j - F_{j'})} \le e^{-(p_j - p_{j'})}$ and so this is at most $e^{-(p_j - p_{j'})}(1 - p_{j'})$, thus proving (R2). □

Proposition 5.7. *Suppose we run* ROUNDBUNDLES(y, z) *for $z \in [0, 1]$. For any client j, then we have:*

1. *If $p_j < z$, then $T_j \le r_j$ with probability at least $1 - e^{-p_j}$;*
2. *If $p_j \ge z$, then $T_j \le 3r_j$ with probability at least $1 - e^{z - p_j}(1 - z)$*

Proof. If $p_j < z$ then by (R1), we have $P(F_j \cap \mathcal{S} \neq \emptyset) \ge 1 - e^{-p_j}$.

Otherwise suppose $p_j \ge z$, so $\sigma(j)$ is defined. Now note that all facilities in $F_j \cup F_{\sigma(j)}$ have distance at most $3r_j$ from facility j (as $r_{\sigma(j)} \le r_j$). By (R2), we have $\Pr[\mathcal{S} \cap (F_j \cup F_{\sigma(j)}) \neq \emptyset] \ge 1 - e^{-p_j + p_{\sigma(j)}}(1 - p_{\sigma(j)})$. As $p_{\sigma(j)} \ge z$, this is is $\ge 1 - e^{-p_j + z}(1 - z)$. So we have $T_j \le 3r_j$ with probability at least $1 - e^{z - p_j}(1 - z)$. □

Corollary 5.8. *There is a* $(3, 0.743)$*-approximation algorithm for CMkC.*

Proof. Apply Proposition 5.7 with $z = 0.62672$.

Suppose $p_j < z$. Then we have $T_j \leq 3r_j$ with probability at least $1 - e^{-p_j} = p_j \frac{1-e^{-p_j}}{p_j} \geq p_j \frac{1-e^{-z}}{z} \geq 0.743 p_j$.

Suppose $p_j \geq z$. Then we have $T_j \leq 3r_j$ with probability at least $1 - e^{-p_j+z}(1-z) = p_j \frac{1-e^{-p_j+z}(1-z)}{p_j} \geq p_j \frac{1-e^{-1+z}(1-z)}{\geq} 0.743 p_j$. □

By choosing z randomly we can achieve a better guarantee

Theorem 5.9. *There is a* $(3, 0.8039)$*-approximation algorithm for CMkC.*

Proof. Run ROUNDBUNDLES(y, Z), where Z is a random variable with pdf $f(z)$, to be defined later. Now, conditioned on $Z = z$, we have $T_j \leq 3r_j$ with probability at least $1 - e^{-p_j}$ for $p_j < z$ and at least $1 - e^{z-p_j}(1-z)$ for $p_j \geq z$. Integrating over $z \in [0, 1]$, we have:

$$
\begin{aligned}
P(T_j > 3r_j) &\leq \int_0^{p_j} f(z) e^{-(p_j - z)}(1-z) dz - \int_{p_j}^1 f(z) e^{-p_j} dz \\
&= e^{-p_j} \left(\int_0^{p_j} f(z) e^z (1-z) dz + 1 - \int_0^{p_j} f(z) dz \right) \\
&= e^{-p_j} \left(1 - \int_0^{p_j} f(z)(1 - e^z(1-z)) dz \right)
\end{aligned}
$$

Now define pdf $f(z) := \frac{e^z((z+1)(1-e^{-z_0})/z_0 - 1)}{1-e^z(1-z)}$ if $z \in [z_0, 1]$ and 0 otherwise, where z_0 is the unique value in $(0, 1]$ such that $\int_{z_0}^1 f(z) dz = 1$. One can show that the numerator and denominator of f are both positive when $z \geq z_0 > 0$, so $f(z) \geq 0$ and thus $f(z)$ is a valid pdf. We find numerically that $0.4534 < z_0 < 0.4535$.

Substituting the formula for f into the expression above, we have that $P(T_j \leq 3r_j) \geq \frac{1-e^{-z_0}}{z_0} p_j$ if $p_j \geq z_0$, and $P(T_j \leq 3r_j) \geq 1 - e^{-p_j}$ if $p_j \leq z_0$. In the former case, the chance factor is at least $\frac{1-e^{z_0}}{z_0} > 0.8039$. In the later case, it is at least $\frac{1-e^{p_j}}{p_j} \geq \frac{1-e^{z_0}}{z_0} > 0.8039$.

□

6 Acknowledgment

We thank Jason Shuster at NASA for the proof of Proposition A.3.

References

[1] FATML: Fairness, Accountability, and Transparency in Machine Learning; http://www.fatml.org/index.html.

[2] Group Buying, Wikipedia: https://en.wikipedia.org/wiki/Group_buying.

[3] A. Ageev and M. Sviridenko. Pipage rounding: a new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8(3):307–328, 2004.

[4] I. Ayres, M. Banaji, and C. Jolls. Race effects on eBay. *The RAND Journal of Economics*, 46(4):891–917, 2015.

[5] E. Badger. How Airbnb plans to fix its racial-bias problem. *Washington Post*, 2016. September 8, 2016. https://www.washingtonpost.com/news/wonk/wp/2016/09/08/how-airbnb-plans-to-fix-its-racial-bias-problem/.

[6] M. Bertrand and S. Mullainathan. Are Emily and Greg More Employable Than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination. *American Economic Review*, 94(4):991–1013, 2004.

[7] D. M. Butler and D. E. Broockman. Do Politicians Racially Discriminate Against Constituents? A Field Experiment on State Legislators. *American Journal of Political Science*, 55(3):463–477, 2011.

[8] J. Byrka, T. Pensyl, B. Rybicki, A. Srinivasan, and K. Trinh. An improved approximation for k-median, and positive correlation in budgeted optimization. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA 2015)*, pages 737–756, 2015.

[9] J. Byrka, A. Srinivasan, and C. Swamy. Fault-tolerant facility location: A randomized dependent lp-rounding algorithm. In *Integer Programming and Combinatorial Optimization, 14th International Conference, IPCO 2010, Lausanne, Switzerland, June 9-11, 2010. Proceedings*, pages 244–257, 2010.

[10] G. Călinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011.

[11] M. Charikar and S. Li. A dependent LP-rounding approach for the k-median problem. *Automata, Languages, and Programming (ICALP)*, pages 194–205, 2012.

[12] C. Chekuri, J. Vondrák, and R. Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *51th Annual IEEE Symposium on Foundations of Computer Science*, pages 575–584, 2010.

[13] C. Chekuri, J. Vondrák, and R. Zenklusen. Multi-budgeted matchings and matroid intersection via dependent rounding. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1080–1097, 2011.

[14] D. Z. Chen, J. Li, H. Liang, and H. Wang. Matroid and Knapsack Center Problems. In *Integer Programming and Combinatorial Optimization: 16th International Conference, IPCO 2013, Valparaíso, Chile, March 18-20, 2013. Proceedings*, pages 110–122. Springer: Berlin, Heidelberg, 2013.

[15] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM*, 53:324–360, 2006.

[16] D. S. Hochbaum and D. B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *J. ACM*, 33(3):533–550, 1986.

[17] S. Khuller, R. Pless, and Y. J. Sussmann. Fault tolerant k-center problems. *Theor. Comput. Sci.*, 242(1-2):237–245, 2000.

[18] J. M. Kleinberg, Y. Rabani, and É. Tardos. Allocating bandwidth for bursty connections. *SIAM J. Comput.*, 30(1):191–217, 2000.

[19] J. M. Kleinberg, Y. Rabani, and É. Tardos. Fairness in routing and load balancing. *J. Comput. Syst. Sci.*, 63(1):2–20, 2001.

[20] S. Li and O. Svensson. Approximating k-median via pseudo-approximation. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 901–910, 2013.

[21] K. A. Schulman, J. A. Berlin, W. Harless, J. F. Kerner, S. Sistrunk, B. J. Gersh, R. Dubé, C. K. Taleghani, J. E. Burke, S. Williams, J. M. Eisenberg, W. Ayers, and J. J. Escarce. The effect of race and sex on physicians' recommendations for cardiac catheterization. *New England Journal of Medicine*, 340(8):618–626, 1999.

[22] A. Srinivasan. Distributions on level-sets with applications to approximation algorithms. In *IEEE Symposium on Foundations of Computer Science*, pages 588–597, 2001.

A Negative correlation in symmetric randomized dependent rounding

The following proposition introduces notation and preliminary analysis used for proving the lower and upper bounds on near-independence.

Proposition A.1. *Let $X = \textsc{Round}(x, a)$ for vectors $x \in (0,1)^n$ and $a \in (\mathbb{R} \setminus \{0\})^n$. For $S \subseteq [n]$ and $c \geq 0$, define*

$$\Lambda := \prod_{i \in S} X_i^p, \qquad \lambda := \prod_{i \in S} x_i^p, \qquad y_i := \begin{cases} \frac{\delta}{a_i x_i} & i \in S \\ 0 & i \notin S \end{cases}, \qquad A_i := (1+y_i)^p, \qquad B_i := (1-y_i)^p.$$

Then

$$\mathbf{E}[\Lambda | (i,j)] := \mathbf{E}[\Lambda | \{i^*, j^*\} = \{i,j\}] = \frac{1}{2}\lambda(A_i B_j + B_i A_j).$$

(Note: The variables defined above are functions of S and/or p, but we omit this to avoid notational clutter.)

Proof. First suppose $i, j \in S$. Because X and x differ only in elements i^* and j^*, we may express

$$\begin{aligned} \mathbf{E}[\Lambda | \{i^*, j^*\} = \{i,j\}] &= \mathbf{E}\left[\lambda \cdot \frac{X_i^p}{x_i^p} \cdot \frac{X_j^p}{x_j^p} \,\middle|\, (i,j)\right] \\ &= \frac{1}{2}\lambda \cdot \frac{(x_i + \delta/a_i)^p}{x_i^p} \cdot \frac{(x_j - \delta/a_j)^p}{x_j^p} + \frac{1}{2}\lambda \cdot \frac{(x_i - \delta/a_i)^p}{x_i^p} \cdot \frac{(x_j + \delta/a_j)^p}{x_j^p} \\ &= \frac{1}{2}\lambda\big((1+y_i)^p(1-y_j)^p + (1-y_i)^p(1+y_j)^p\big) \qquad (13) \\ &= \frac{1}{2}\lambda(A_i B_j + B_i A_j) \end{aligned}$$

Lastly, observe equation (13) is still true even when i or j are not in S. □

A.1 Upper bound on near-independence

Here we will prove the upper bound of Theorem 2.2. We start with a technical lemma. Note the binomial coefficient $\binom{p}{k}$ is defined for real $p \geq 0$ and integer $k \geq 0$ to be:

$$\binom{p}{k} := \frac{\prod_{i=0}^{k-1}(p-i)}{k!}.$$

Proposition A.2. *For all $x \in [-1,1]$, $n \geq 2$, and $p = 1 - \frac{1}{n}$,*

$$n(1+x)^{2p} + n(1-x)^{2p} + 2(n-2)(1-x^2)^p \leq 4(n-1).$$

Proof. First expand the LHS using the generalized binomial theorem:

$$
\begin{aligned}
&n(1+x)^{2p} + n(1-x)^{2p} + 2(n-2)(1-x^2)^p \\
=&n\sum_{k\geq 0}\binom{2p}{k}x^k + n\sum_{k\geq 0}\binom{2p}{k}(-1)^k x^k + 2(n-2)\sum_{k\geq 0}\binom{p}{k}(-1)^k x^{2k} \\
=&2n\sum_{\ell\geq 0}\binom{2p}{2\ell}x^{2\ell} + 2(n-2)\sum_{k\geq 0}\binom{p}{k}(-1)^k x^{2k} \\
=&2\sum_{k\geq 0}\Big(n\binom{2p}{2k} + (n-2)\binom{p}{k}(-1)^k\Big)x^{2k}. \qquad (14)
\end{aligned}
$$

Now define $f_k(n) := n\binom{2p}{2k} + (n-2)\binom{p}{k}(-1)^k$ for $k = 0, 1, 2\ldots$ where $p = 1 - 1/n$. Then

$$
\begin{aligned}
f_0(n) &= n\cdot 1 + (n-2)\cdot 1\cdot 1 = 2(n-1), \\
f_1(n) &= n\frac{2p(2p-1)}{2} - (n-2)p = 2p\big((p-1)n+1\big) = 2p\Big(-\frac{1}{n}n+1\Big) = 0.
\end{aligned}
$$

We claim the following recurrence holds, which can be shown algebraically using $\frac{n-k+1}{k}\binom{n}{k-1} = \binom{n}{k}$:

$$
f_k(n) = \Big(1 - \frac{p+1}{k}\Big)\Big(f_{k-1}(n,p) - n\frac{2p}{2k-1}\binom{2p}{2(k-1)}\Big).
$$

When $k \geq 2$, $n \geq 2$, and $p = (1-\frac{1}{n})$, we have $(1-\frac{p+1}{k}) \in (0,1)$, $\binom{2p}{2(k-1)} \geq 0$, and $n\frac{2p}{2k-1} > 0$. Thus the recurrence implies $f_k(n) < 1\cdot(f_{k-1}(n) - 0\cdot 0) = f_{k-1}(n) < \ldots < f_1(n) = 0$. Continuing from (14),

$$
n(1+x)^{2p} + n(1-x)^{2p} + 2(n-2)(1-x^2)^p = 2\sum_{k\geq 0} f_k(n)x^{2k} \leq 2f_0(n) = 4(n-1).
$$

$\square$

Proposition 2.3. *Let $X = \textsc{Round}(x,a)$ for vectors $x \in (0,1)^n$ and $a \in (\mathbb{R}\setminus\{0\})^n$, $n \geq 2$. For any set $S \subseteq [n]$,*

$$
\mathbf{E}\Big[\Big(\prod_{i\in S} X_i\Big)^p\Big] \leq \Big(\prod_{i\in S} x_i\Big)^p \qquad (15)
$$

holds for $p = 1 - \frac{1}{n}$. Furthermore, if all weights in a_S have the same sign, the inequality holds for $p = 1$.

Proof. Fix set $S \in [n]$. First consider the case that all weights in a_S have the same sign. Then all values in y_S also have the same sign or are zero. Applying Proposition A.1 with $p = 1$, we see

$$
\mathbf{E}[\Lambda|(i,j)] = \frac{1}{2}\lambda(A_iB_j + B_iA_j) = \frac{1}{2}\lambda\big((1+y_i)(1-y_j) + (1-y_i)(1+y_j)\big) = \lambda(1-y_iy_j) \leq \lambda.
$$

Since this is true for all pairs (i,j), it holds that $\mathbf{E}[\Lambda] \leq \lambda$, which is equivalent to (15).

We now consider the general case. In the following summations, $i, j \in [n]$.

$$\begin{aligned}
\mathbf{E}[\Lambda] =& \sum_{i<j} \Pr[\{i^*, j^*\} = \{i, j\}]\, \mathbf{E}[\Lambda|(i,j)] \\
=& \sum_{i<j} \frac{1}{\binom{n}{2}} \cdot \frac{1}{2}\lambda(A_iB_j + B_iA_j) \\
=& \frac{\lambda}{2\binom{n}{2}} \cdot \frac{1}{2} \sum_{i<j} (A_i+B_i)(A_j+B_j) - (A_i-B_i)(A_j-B_j) \\
=& \frac{\lambda}{4\binom{n}{2}} \cdot \frac{1}{2} \Big(\Big(\sum_i (A_i+B_i) \Big)^2 - \sum_i (A_i+B_i)^2 - \Big(\sum_i (A_i-B_i) \Big)^2 + \sum_i (A_i-B_i)^2 \Big) \\
\leq& \frac{\lambda}{8\binom{n}{2}} \Big(n \sum_i (A_i+B_i)^2 - \sum_i (A_i+B_i)^2 + \sum_i (A_i-B_i)^2 \Big) \qquad (16) \\
\mathbf{E}[\Lambda] =& \frac{\lambda}{8\binom{n}{2}} \sum_i \big(n(A_i^2 + B_i^2) + 2(n-2)A_iB_i \big).
\end{aligned}$$

To get (16) we applied the Cauchy-Schwarz inequality and the nonnegativity of squares. Now expand A_i and B_i. By observing that $y_i \in [0,1]$ and fixing $p = 1 - \frac{1}{n}$, we may apply Proposition A.2:

$$\begin{aligned}
\mathbf{E}[\Lambda] &= \frac{\lambda}{8\binom{n}{2}} \sum_i \big(n(1+y_i)^{2p} + n(1-y_i)^{2p} + 2(n-2)n(1-y_i^2)^{2p} \big) \\
&\leq \frac{\lambda}{8\binom{n}{2}} \sum_i 4(n-1) = \frac{\lambda}{8\binom{n}{2}} 4n(n-1) = \lambda.
\end{aligned}$$

□

A.2 Lower bound on near-independence

Here we will prove the lower bound of Theorem 2.2. We start with a technical lemma.

Proposition A.3. *For all $x \in [-1,1]$, $n \geq 2$, $1 \leq s < n$, and $p \geq 1 + \frac{s-1}{n-s} \geq 1$,*

$$(n-s)\big((1+x)^p + (1-x)^p\big) + (s-1)(1-x^2)^p \geq 2n - s - 1. \qquad (17)$$

Proof. Define

$$f(x) := (n-s)\big((1+x)^p + (1-x)^p\big) + (s-1)(1-x^2)^p - (2n-s-1).$$

We wish to show $f(x) \geq 0$. Observe $f(0) = 2(n-s) + (s-1) - 2n + s + 1 = 0$. Since f is continuous, it thus suffices to show $f'(x) \geq 0$ for $x \in [0,1)$.

$$f'(x) = p(n-s)(1+x)^{p-1} - p(n-s)(1-x)^{p-1} - 2xp(s-1)(1-x^2)^{p-1}.$$

When $x \in [0,1)$, the quantity $p(1+x)^{p-1}(1-x)^{p-1} > 0$. Thus it is equivalent to show $g(x) \geq 0$, where

$$g(x) := \frac{f'(x)}{p(1-x)^{p-1}(1+x)^{p-1}} = (n-s)(1-x)^{1-p} - (n-s)(1+x)^{1-p} - 2x(s-1).$$

Again, since $g(0) = 0$ and g is continuous (for $x \in [0, 1)$), it suffices to show that $g'(x) \geq 0$ for $x \in [0, 1)$. We do this using Jensen's inequality, since x^{-p} is convex when $p \geq 0$.

$$\begin{aligned}
g'(x) &= -(1-p)(n-s)(1-x)^{-p} - (1-p)(n-s)(1+x)^{-p} - 2(s-1) \\
&= (p-1)(n-s)\Big((1-x)^{-p} + (1+x)^{-p}\Big) - 2(s-1) \\
&\geq (p-1)(n-s) \cdot 2 - 2(s-1) \\
&\geq (\frac{s-1}{n-s})(n-s) \cdot 2 - 2(s-1) = 0.
\end{aligned}$$

$\square$

Thanks to Jason Shuster at NASA for the above proof.

Proposition 2.5. *Let $X = \textsc{Round}(x, a)$ for vectors $x \in (0,1)^n$ and $a \in (\mathbb{R} \setminus \{0\})^n$, $n \geq 2$. For set $S \subseteq [n]$, define $S_1 := \{i \in S \mid a_i > 0\}$ and $S_2 := \{i \in S \mid a_i < 0\}$. If $|S_1|, |S_2| \leq n-1$, then*

$$\mathbf{E}\left[\Big(\prod_{i \in S} X_i\Big)^p\right] \geq \Big(\prod_{i \in S} x_i\Big)^p$$

holds for $p = 1 + \max\left\{\frac{|S_1|-1}{n-|S_1|}, \frac{|S_2|-1}{n-|S_2|}\right\}$.

Proof. Using the notation of Proposition A.1, we need to show $\mathbf{E}[\Lambda] \geq \mathbf{E}[\lambda]$, or equivalently (since λ is constant), $\mathbf{E}[\frac{\Lambda}{\lambda}] \geq 1$. We decompose $\mathbf{E}[\frac{\Lambda}{\lambda}]$ into a sum of expectations conditional on (i, j).

$$\mathbf{E}\left[\frac{\Lambda}{\lambda}\right] = \sum_{i<j} \Pr\Big[\{i^*, j^*\} = \{i, j\}\Big]\, \mathbf{E}\left[\frac{\Lambda}{\lambda} \,\Big|\, (i,j)\right] = \frac{1}{\binom{n}{2}} \sum_{i<j} \mathbf{E}\left[\frac{\Lambda}{\lambda} \,\Big|\, (i,j)\right] \tag{18}$$

We will partition the conditional expectations according to the signs of y_i and y_j.

Case $\boldsymbol{y_i y_j < 0}$. Notice $A_i > B_i$ when $y_i > 0$ and $A_i < B_i$ when $y_i < 0$. Since y_i and y_j are opposite signs: $-(A_i - B_i)(A_j - B_j) > 0$. Also observe for $x, y \geq 2$, it is true that $xy = 2x + 2y - 4 + (x-2)(y-2) \geq 2x + 2y - 4$. By Bernoulli's inequality, $A_i + B_i = (1+y_i)^p + (1-y_i)^p \geq 1 + py_i + 1 - py_i = 2$. Therefore $(A_i + B_i)(A_j + B_j) \geq 2(A_i + B_i) + 2(A_j + B_j) - 4$. Altogether we get

$$\begin{aligned}
\mathbf{E}\left[\frac{\Lambda}{\lambda} \,\Big|\, (i,j) \wedge (y_i y_j < 0)\right] &= \frac{1}{\lambda} \cdot \frac{1}{2}\lambda(A_i B_j + B_i A_j) \\
&= \frac{1}{4}\big((A_i + B_i)(A_j + B_j) - (A_i - B_i)(A_j - B_j)\big) \\
&> \frac{1}{2}(A_i + B_i) + \frac{1}{2}(A_j + B_j) - 1.
\end{aligned} \tag{19}$$

Now sum (19) over all pairs $(i, j) \in S_1 \times S_2$ (i.e. those pairs such that $y_i y_j < 0$.) Define $\bar{y}_1 := \frac{1}{s_1}\sum_{i \in S_1} y_i$ and $\bar{y}_2 := \frac{1}{s_2}\sum_{i \in S_2} y_i$. Also define $s_1 := |S_1|$, $s_2 := |S_2|$. We observe the function x^p is convex (for $p \geq 1$)

and apply Jensen's inequality to get a quantity in terms of $\bar{y}_1$ and $\bar{y}_2$:

$$\begin{aligned}
\sum_{\substack{i<j \\ y_iy_j<0}} \mathbf{E}\left[\frac{\Lambda}{\lambda} \,\Big|\, (i,j)\right] &> \sum_{i \in S_1} \sum_{j \in S_2} \Big(\frac{1}{2}(A_i + B_i) + \frac{1}{2}(A_j + B_j) - 1\Big) \\
&= \frac{s_2}{2} \sum_{i \in S_1} (A_i + B_i) + \frac{s_1}{2} \sum_{i \in S_2} (A_i + B_i) - s_1 s_2 \\
&= \frac{s_2}{2} \sum_{i \in S_1} ((1+y_i)^p + (1-y_i)^p) + \frac{s_1}{2} \sum_{i \in S_2} ((1+y_i)^p + (1-y_i)^p) - s_1 s_2 \\
&\geq \frac{s_2}{2} s_1((1+\bar{y}_1)^p + (1-\bar{y}_1)^p) + \frac{s_1}{2} s_2((1+\bar{y}_2)^p + (1-\bar{y}_2)^p) - s_1 s_2 \\
&= \frac{s_1 s_2}{2} \big((1+\bar{y}_1)^p + (1-\bar{y}_1)^p + (1+\bar{y}_2)^p + (1-\bar{y}_2)^p - 2\big) \qquad (20)
\end{aligned}$$

Case $y_iy_j > 0$. Applying Jensen's inequality for x^p, we get

$$\begin{aligned}
\mathbf{E}\left[\frac{\Lambda}{\lambda} \,\Big|\, (i,j) \wedge (y_iy_j > 0)\right] &= \frac{1}{\lambda} \cdot \frac{1}{2} \lambda (A_i B_j + B_i A_j) \\
&= \frac{1}{2}(1+y_i)^p(1-y_j)^p + \frac{1}{2}(1-y_i)^p(1+y_j)^p \\
&= \frac{1}{2}(1 + y_i - y_j - y_iy_j)^p + \frac{1}{2}(1 - y_i + y_j - y_iy_j)^p \\
&\geq (1 - y_iy_j)^p \qquad (21)
\end{aligned}$$

Now sum (21) over all pairs (i,j) where $y_iy_j > 0$, that is, where either $i,j \in S_1$ or $i,j \in S_2$. Maclaurin's inequality states that $\frac{1}{\binom{m}{2}} \sum_{1 \leq i<j \leq m}^{m} y_iy_j \leq (1/n \sum_{i=1}^{m} y_i)^2$. This together with Jensen's gives:

$$\begin{aligned}
\sum_{\substack{i<j \\ y_iy_j>0}} \mathbf{E}\left[\frac{\Lambda}{\lambda} \,\Big|\, (i,j)\right] &= \sum_{\substack{i<j \\ i,j \in S_1}} \mathbf{E}\left[\frac{\Lambda}{\lambda} \,\Big|\, (i,j)\right] + \sum_{\substack{i<j \\ i,j \in S_2}} \mathbf{E}\left[\frac{\Lambda}{\lambda} \,\Big|\, (i,j)\right] \\
&\geq \sum_{\substack{i<j \\ i,j \in S_1}} (1 - y_iy_j)^p + \sum_{\substack{i<j \\ i,j \in S_2}} (1 - y_iy_j)^p \\
&\geq \binom{s_1}{2} \Bigg(1 - \frac{1}{\binom{s_1}{2}} \sum_{\substack{i<j \\ i,j \in S_1}} y_iy_j\Bigg)^p + \binom{s_2}{2} \Bigg(1 - \frac{1}{\binom{s_2}{2}} \sum_{\substack{i<j \\ i,j \in S_2}} y_iy_j\Bigg)^p \\
&\geq \binom{s_1}{2} (1 - \bar{y}_1^2)^p + \binom{s_2}{2} (1 - \bar{y}_2^2)^p \qquad (22)
\end{aligned}$$

Case $y_iy_j = 0$. Define $S_3 := [n] \setminus S$. Then $y_i = 0 \iff i \in S_3$. When y_i is 0, then $A_i = B_i = 1$, so $\mathbf{E}[\Lambda | (i,j) \wedge (y_i = 0)] = \frac{1}{2}\lambda(A_j + B_j)$. If $y_i = y_j = 0$, then $\mathbf{E}[\Lambda | (i,j) \wedge (y_i = y_j = 0)] = \lambda$. Again we

use Jensen's:

$$\sum_{\substack{i<j\\ y_iy_j=0}} \mathbf{E}\left[\frac{\Lambda}{\lambda}\,\middle|\,(i,j)\right] = \sum_{i\in S_3}\sum_{j\in S_1} \mathbf{E}\left[\frac{\Lambda}{\lambda}\,\middle|\,(i,j)\right] + \sum_{i\in S_3}\sum_{j\in S_2} \mathbf{E}\left[\frac{\Lambda}{\lambda}\,\middle|\,(i,j)\right] + \sum_{\substack{i<j\\ i,j\in S_3}} \mathbf{E}\left[\frac{\Lambda}{\lambda}\,\middle|\,(i,j)\right]$$
$$= \sum_{i\in S_3}\sum_{j\in S_1} \frac{1}{2}((1+y_j)^p + (1-y_j)^p) + \sum_{i\in S_3}\sum_{j\in S_2} \frac{1}{2}((1+y_j)^p + (1-y_j)^p) + \sum_{\substack{i<j\\ i,j\in S_3}} 1$$
$$\geq \sum_{i\in S_3} \frac{s_1}{2}((1+\bar{y}_1)^p + (1-\bar{y}_1)^p) + \sum_{i\in S_3} \frac{s_2}{2}((1+\bar{y}_2)^p + (1-\bar{y}_2)^p) + \binom{n-s_1-s_2}{2}$$
$$= \frac{n-s_1-s_2}{2}\Big(s_1\big((1+\bar{y}_1)^p + (1-\bar{y}_1)^p\big) + s_2\big((1+\bar{y}_2)^p + (1-\bar{y}_2)^p\big) + n-s_1-s_2-1\Big). \tag{23}$$

Combining (18), (20), (22), and (23), we can bound $E[\frac{\Lambda}{\lambda}]$:

$$\binom{n}{2}\mathbf{E}\left[\frac{\Lambda}{\lambda}\right] = \sum_{\substack{i<j\\ y_iy_j=0}} \mathbf{E}\left[\frac{\Lambda}{\lambda}\,\middle|\,(i,j)\right] + \sum_{\substack{i<j\\ y_iy_j=0}} \mathbf{E}\left[\frac{\Lambda}{\lambda}\,\middle|\,(i,j)\right] + \sum_{\substack{i<j\\ y_iy_j=0}} \mathbf{E}\left[\frac{\Lambda}{\lambda}\,\middle|\,(i,j)\right]$$
$$\geq \frac{s_1s_2}{2}\big((1+\bar{y}_1)^p + (1-\bar{y}_1)^p + (1+\bar{y}_2)^p + (1-\bar{y}_2)^p - 2\big) + \binom{s_1}{2}(1-\bar{y}_1^2)^p + \binom{s_2}{2}(1-\bar{y}_2^2)^p$$
$$+ \frac{n-s_1-s_2}{2}\Big(s_1\big((1+\bar{y}_1)^p + (1-\bar{y}_1)^p\big) + s_2\big((1+\bar{y}_2)^p + (1-\bar{y}_2)^p\big) + n-s_1-s_2-1\Big)$$
$$= \frac{s_1(n-s_1)}{2}\Big((1+\bar{y}_1)^p + (1-\bar{y}_1)^p\Big) + \binom{s_1}{2}(1-\bar{y}_1^2)^p$$
$$+ \frac{s_2(n-s_2)}{2}\Big((1+\bar{y}_2)^p + (1-\bar{y}_2)^p\Big) + \binom{s_2}{2}(1-\bar{y}_2^2)^p - s_1s_2 + \binom{n-s_1-s_2}{2}. \tag{24}$$

Finally, we'd like to apply Proposition A.3 separately for $\bar{y}_1$ and $\bar{y}_2$. Fix $p = 1 + \max\left\{\frac{s_1-1}{n-s_1}, \frac{s_2-1}{n-s_2}\right\}$. Observe that the means $\bar{y}_1, \bar{y}_2 \in (0,1)$, and recall that $s_1, s_2 \leq n-1$ and $n \geq 2$ are given. If we further assume that $s_1, s_2 \geq 1$, then all the necessary conditions for Proposition A.3 hold and we may apply it twice to (24) to get:

$$\binom{n}{2}\mathbf{E}\left[\frac{\Lambda}{\lambda}\right] \geq \frac{s_1}{2}(2n-s_1-1) + \frac{s_2}{2}(2n-s_2-1) - s_1s_2 + \binom{n-s_1-s_2}{2}$$
$$= \frac{1}{2}(2n-1)(s_1+s_2) - \frac{1}{2}(s_1+s_2)^2 + \frac{1}{2}(n-(s_1+s_2))(n-1-(s_1+s_2))$$
$$= \frac{n(n-1)}{2} = \binom{n}{2}.$$

Else at least one of s_1 and s_2 is 0. If both are zero, the LHS of (24) becomes $\binom{n}{2}$. If just one is zero (s_1 for example), then we apply Proposition A.3 only once:

$$\binom{n}{2}\mathbf{E}\left[\frac{\Lambda}{\lambda}\right] \geq \frac{s_2(n-s_2)}{2}\Big((1+\bar{y}_2)^p + (1-\bar{y}_2)^p\Big) + \binom{s_2}{2}(1-\bar{y}_2^2)^p + \binom{n-s_2}{2}.$$
$$\geq \frac{s_2}{2}(2n-s_2-1) + \frac{1}{2}(n-s_2)(n-s_2-1) = \binom{n}{2}.$$

□

B Omitted proofs for Section 3

For each $j \in W$, let us define the random variable $\mu_j := \sum_{i \in C_j} \delta_i$. We also define the random variable U_j as follows:

1. We set $U_j = 0$ if $j \notin J$.
2. Otherwise, if $i_{j1}, i_{j2} \in C_j$ or $i_{j1}, i_{j2} \notin C_j$, we set $U_j = 0$.
3. Otherwise, we set $U_j = \min(y_{i_{j1}}, y_{i_{j2}})$.

Observe that the random variables U_j are independent (as each only depends on the choices made within G_j). The random variables μ_j are highly interdependent.

Proposition B.1. *For any $j \in W$, we have $|\mu_j| \leq U_j$.*

Proof. If $j \notin J$, then necessarily $\mu_j = 0 = U_j$ and we are done. So suppose $j \in J$. By construction and our choice of δ, if $i_{j1}, i_{j2} \in C_j$ or $i_{j1}, i_{j2} \notin C_j$ then $\mu_j = 0 = U_j$.

Finally, observe that we must have $|\delta_{i_{j1}}| \leq y_{i_{j1}}$ as $y \pm \delta \in [0,1]^n$. Similarly, $|\delta_{i_{j2}}| \leq y_{i_{j2}}$. As $|\delta_{i_{j1}}| = |\delta_{i_{j2}}|$ and either $\mu_j = \delta_{i_{j1}}$ or $\mu_j = \delta_{i_{j2}}$, it follows that $|\mu_j| \leq \min(y_{i_{j1}}, y_{i_{j2}})$ as we have claimed. $\square$

Proposition B.2. *For any $j \in W$, we have*

$$\mathbf{E}[U_j] \leq p_j \frac{2c_j(1-c_j)}{|F_j| - 1}.$$

Proof. First, observe that $j \in J$ with probability p_j, and this is independent of i_{j1}, i_{j2}.

Suppose we enumerate $C_j = \{1, \dots, l\}$ and $G_j - C_j = \{l+1, \dots, r\}$ where $r = |G_j|$. As i_{j1}, i_{j2} are chosen uniformly at random, we have

$$\mathbf{E}[U_j] = p_j \frac{\sum_{u=1}^{l} \sum_{v=l+1}^{r} \min(y_u, y_v)}{\binom{r}{2}}. \tag{25}$$

Now, we claim that for a fixed value of the sum $c_j = y_1 + \dots + y_l$ and $y_{l+1} + \dots + y_r = 1 - c_j$, the RHS of (25) is maximized when $y_1 = y_2 = \dots = y_l = c_j/l$ and $y_{l+1} = \dots = y_r = (1-c_j)/r$. For, suppose (WLOG) that $l > 1$ and $y_1 > y_2$ and we decrement y_1 by some small $x > 0$ and increment y_2 by x. We choose x sufficiently small so that we have

$$y_1 > y_v > y_2 \qquad \text{iff} \qquad y_1 - x > y_v > y_2 + x$$

for $v = l+1, \dots, r$. Then $\mathbf{E}[U_j]$ changes by $\sum_{v \in [l+1,r]: y_1 > y_e > y_2} x > 0$.

Thus,

$$\mathbf{E}[U_j] \leq p_j \frac{\sum_{u=1}^{l} \sum_{v=l+1}^{r} \min(c_j/l, (1-c_j)/(r-l))}{\binom{r}{2}} = p_j \frac{l(r-l)\min(c_j/l, (1-c_j)/(r-l))}{\binom{r}{2}}$$

Now, suppose that we ignore the constraint that l is an integer in $[r]$ and relax this to the weaker condition that $l \in [0, r]$. We then have

$$\mathbf{E}[U_j] \leq p_j \frac{\max_{l \in [0,r]} T(l)}{\binom{r}{2}} \qquad \text{where} \qquad T(l) = l(r-l)\min(\frac{c_j}{l}, \frac{1-c_j}{r-l})$$

We claim that the maximum value of $T(l)$ can only occur when $c_j/l = (1-c_j)/(r-l)$. For, suppose that $c_j/l < (1-c_j)/(r-l)$. Then, within an open ball of l, we have $T(l) = (r-l)c_j$, and hence

$T'(l) = -c_j \neq 0$. Similarly, if $c_j/l > (1-c_j)/(r-l)$, we have $T(l) = l(1-c_j)$ and $T'(l) = -c_j \neq 0$. Thus, we have that $T(l)$ is maximized at $l = c_j r$ yielding $T(l) \leq rc_j(1-c_j)$ and hence

$$\mathbf{E}[U_j] \leq p_j \frac{2c_j(1-c_j)}{r-1}.$$

□

Proposition B.3. *Suppose we condition on all the state at the beginning of the dependent rounding round q. Then*

$$\mathbf{E}[S'] \leq S \cosh\left(\sum_{j \in W} \frac{\mathbf{E}[U_j]}{1-c_j}\right).$$

Proof. Let us first condition on the random variable δ (which includes all the random choices up to but not including line 9 in Algorithm 7). With probability $1/2$ we change y to $y+\delta$ and with probability $1/2$ we change y to $y-\delta$. Thus,

$$\begin{aligned}
\mathbf{E}[S' \mid \delta] &= 1/2 \prod_{j \in W}(1-c_j-\mu_j) + 1/2 \prod_{j \in W}(1-c_j+\mu_j) \\
&= 1/2 \sum_{X \subseteq W} \prod_{j \in X}(-\mu_j) \prod_{j \in W-X}(1-c_j) + 1/2 \sum_{X \subseteq W} \prod_{j \in X}(\mu_j) \prod_{j \in W-X}(1-c_j) \\
&= \sum_{\substack{X \subseteq W \\ |X| \text{ even}}} \prod_{j \in X} \mu_j \prod_{j \in W-X}(1-c_j) \\
&= S \sum_{\substack{X \subseteq W \\ |X| \text{ even}}} \prod_{j \in X} \frac{\mu_j}{1-c_j} \qquad (\text{as } S = \prod_{j \in W}(1-c_j)) \\
&\leq S \sum_{\substack{X \subseteq W \\ |X| \text{ even}}} \prod_{j \in X} \frac{|\mu_j|}{1-c_j} \\
&\leq S \sum_{\substack{X \subseteq W \\ |X| \text{ even}}} \prod_{j \in X} \frac{U_j}{1-c_j}. \qquad (\text{by Proposition B.1})
\end{aligned}$$

Integrating over the randomness involved in $J, i_{j1}, i_{j2}, \delta$ gives:

$$\begin{aligned}
\mathbf{E}[S'] &\leq S \sum_{\substack{X \subseteq W \\ |X| \text{ even}}} \mathbf{E}\left[\prod_{j \in X} \frac{U_j}{1-c_j}\right] \\
&= S \sum_{\substack{X \subseteq W \\ |X| \text{ even}}} \prod_{j \in X} \frac{\mathbf{E}[U_j]}{1-c_j} \qquad (\text{as } U_k \text{ are independent}) \\
&= S \sum_{v=0}^{\infty} \sum_{\substack{X \subseteq W \\ |X| = 2v}} \prod_{j \in X} \frac{\mathbf{E}[U_j]}{1-c_j} \\
&\leq S \sum_{v=0}^{\infty} \frac{\left(\sum_{j \in W} \frac{\mathbf{E}[U_j]}{1-c_j}\right)^{2v}}{(2v)!} = S \cosh\left(\sum_{j \in W} \frac{\mathbf{E}[U_j]}{1-c_j}\right).
\end{aligned}$$

$\square$

Proposition B.4. *Suppose that $t \geq 6m^2$. Then*

$$\mathbf{E}[S'] \leq S + \frac{18m^2}{\left(\sum_{j \in W} |\text{frac}(G_j)| - 1\right)^2}$$

Proof. We have

$$\begin{aligned}
\mathbf{E}[S'] &\leq S \cosh\left(\sum_{j \in W} \frac{\mathbf{E}[U_j]}{1 - c_j}\right) \qquad \text{(Proposition B.3)} \\
&\leq S \cosh\left(\sum_{j \in W} (3m) \times \frac{|\text{frac}(G_j)| - 1}{\sum_{j \in W} |\text{frac}(G_j)| - 1} \times \frac{2c_j(1 - c_j)}{(|\text{frac}(G_j)| - 1)(1 - c_j)}\right) \qquad \text{(Proposition B.2)} \\
&= S \cosh\left(\frac{6m}{\sum_{j \in W} (|\text{frac}(G_j)| - 1)} \sum_{j \in W} c_j\right) \\
&\leq S \cosh\left(\frac{-6m}{\sum_{j \in W} |\text{frac}(G_j)| - 1} \sum_{j \in W} \ln(1 - c_j)\right) = S + \left[S \cosh\left(\frac{-6m}{\sum_{j \in W} |\text{frac}(G_j)| - 1} \ln S\right) - S\right].
\end{aligned}$$

Now let $z = \frac{6m}{\sum_{j \in W} (|\text{frac}(G_j)| - 1)}$. Observe that $z \leq \frac{6m}{t} \leq 1$. Now, consider the function $f : [0, 1] \to \mathbb{R}$ defined by

$$f(s) = s \cosh(-z \ln s) - s.$$

Simple analysis shows that for $z < 1$, this achieves a maximum value at $s = \left(\frac{1-z}{1+z}\right)^{1/z}$ so that

$$f(s) \leq \frac{2z^2 \left(\frac{2}{z+1} - 1\right)^{\frac{1}{z}}}{1 - z^2} \leq z^2/2.$$

Thus,

$$\mathbf{E}[S'] \leq S + f(S) \leq S + \frac{18m^2}{\left(\sum_{j \in W} |\text{frac}(G_j)| - 1\right)^2}.$$

$\square$

We are now ready to analyze *multiple* rounds of the dependent rounding process.

Proposition 3.19. *Let S denote the value of the potential function at the beginning of the dependent rounding process (immediately after line 1 of* KDR*), and let S' denote the value of the potential function **at the end of all rounds** (i.e. at line 10 of* KDR*, after terminating $\sum_{j \in W} (|\text{frac}(G_j)| - 1) \leq t$. Then*

$$\mathbf{E}[S'] \leq S + \frac{180m^2}{t}.$$

Proof. First, suppose that $t \leq 6m^2$. Then RHS is at least equal to $\frac{180m^2}{6m^2} = 30$. As $S' \in [0, 1]$ with certainty, this result holds vacuously. So we suppose for the remainder of the proof that $t > 6m^2$.

Let S_q denote the potential function after q rounds (so $S_0 = S$) and let $\gamma_q = \sum_{j \in W} (\text{frac}(|G_j|) - 1)$ after q rounds (recall that the vector y, and consequently the fractional clusters $\text{frac}(G_j)$, change after each round.) By Proposition B.4, conditional on the state at the beginning of round q, we have

$$\mathbf{E}[S_{q+1} \mid \text{state at round } q] \leq S_q + \frac{18m^2}{\gamma_q^2}$$

We also have that $S_{q+1} = S_q$ with certainty if $\gamma_q \leq t$, so can write:

$$\mathbf{E}[S_{q+1} \mid \text{state at round } q] \leq S_q + \frac{18m^2[\gamma_q > t]}{\gamma_q^2},$$

and thus, by iterated expectations, we have that, for all $q \geq 0$,

$$\mathbf{E}[S_q] \leq S + \sum_{z=0}^{q-1} \frac{18[\gamma_z > t]m^2}{\gamma_z^2}.$$

We can write this equivalently as:

$$\begin{aligned} \mathbf{E}[S_q] &\leq S + 18m^2 \sum_{u=t+1}^{\infty} \frac{1}{u^2} \mathbf{E}\left[\sum_{z=0}^{q} [\gamma_z = u]\right] \\ &\leq S + 18m^2 \sum_{u=t+1}^{\infty} \frac{1}{u^2} \mathbf{E}\left[\#z : \gamma_z = u\right]. \end{aligned}$$

By Proposition 3.17, there is a probability of at least $1/10$ that there is a rounded variable in any given round, in which case γ_z decreases by at least one. We now claim that for any integer u, we have $\mathbf{E}\left[\#z : \gamma_z = u\right] \leq 10$. For, suppose we condition on that $\gamma_z = u$ for the first time at $z = z_0$. Then, in each subsequent round, there is a probability of at least $1/10$ that γ_z is no longer equal to u. So the expected number of rounds for which $\gamma_z = u$ is dominated by a geometric random variable with probability $1/10$, and hence has mean ≤ 10.

Thus, for any $q \geq 0$ we have:

$$\mathbf{E}[S_q] \leq S + 18m^2 \sum_{u=t+1}^{\infty} \frac{1}{u^2} \mathbf{E}\left[\#z : \gamma_z = u\right] \leq S + 180m^2 \sum_{u=t+1}^{\infty} \frac{1}{u^2} \leq S + \frac{180m^2}{t}$$

With probability one, the dependent rounding process terminates at some finite (random) number of rounds Q, that is, $S' = S_Q$. Observe that $S_Q \in [0,1]$ with certainty, and if $q > Q$ we have that $S_q = S_Q$. Thus, for any $q \geq 0$, we have $S_Q \leq S_q + [Q > q]$. Taking expectation, this implies that

$$\mathbf{E}[S_Q] \leq \mathbf{E}[S_q] + \Pr[Q > q] \leq S + \frac{180m^2}{t} + \Pr[Q > q].$$

Now take the limit of both sides as $q \to \infty$. Observe that Q is finite with probability one so that $\Pr[Q > q] \to 0$, and hence

$$\mathbf{E}[S'] = \mathbf{E}[S_Q] \leq S + \frac{180m^2}{t}.$$

$\square$

C Omitted proofs for Section 4

In this section, we will analyze the k-center algorithm which uses partial clusters.

Recall that for $j \in [n]$, we let $G_j = F_{\pi(j)} - F_{\pi(1)} - \dots - F_{\pi(j-1)}$; we refer to G_j as a *cluster* and $c_j = \pi(j)$ as the *cluster center* of the partial cluster G_j. We let $z_j = y(G_j)$. We say that G_j is a *full cluster* if $z_j = 1$ and a *partial cluster* otherwise. The reader should be careful of our notational convention here: the center of cluster F_j is vertex j itself, while the center of cluter G_j is $\pi(j)$.

We begin our analysis of the partial-cluster by noting a few simple facts about the parameters z.

Proposition C.1. *The parameters $z_1, \dots, z_n$ satisfy the following properties:*

1. $z_1 = 1$,
2. *For any $j < n$, we have $z_j \geq z_{j+1}$,*
3. *For any $j \in [n]$ and $i \in V$, we have $z_j \geq y(F_i - F_{\pi(1)} - \dots - F_{\pi(j-1)})$.*

Proof. To see (1), note that for any $j \in [n]$ we have $y(F_j) = 1$.

To see (3), first note that if $i \in \{\pi(1), \dots, \pi(j-1)\}$ then $y(F_i - F_{\pi(1)} - \dots - F_{\pi(j-1)}) = y(\emptyset) = 0$, whereas $z_j \geq 0$. If $i \notin \{\pi(1), \dots, \pi(j-1)\}$ then this follows as $\pi(j)$ is chosen to maximize $y(F_{\pi(j)} - F_{\pi(1)} - \dots - F_{\pi(j-1)})$.

To see (2), we apply (3) with $i = \pi(j+1)$ to obtain

$$z_j \geq y(F_{\pi(j+1)} - F_{\pi(1)} - \dots - F_{\pi(j-1)}) \geq y(F_{\pi(j+1)} - F_{\pi(1)} - \dots - F_{\pi(j-1)} - F_{\pi(j)}) = z_{j+1}.$$

□

Proposition C.2. *For any $S \subseteq [n]$, we have*

$$\Pr(\mathcal{S} \cap S = \emptyset \mid Q_f, Q_p) \leq \prod_{j=1}^{n} (1 - (1 - q_j) y(S \cap G_j) - q_j z_j [c_j \in S])$$

Proof. The proof is very similar to Proposition 5.5 and is omitted. □

We now introduce our main result for the value $\mathbf{E}[T_i]$ on an arbitrary vertex i.

Proposition C.3. *Let $i \in V$. Define $J_f, J_p \subseteq V$ as*

$$J_f = \{j \in [n] \mid F_i \cap G_j \neq \emptyset, z_j = 1\}$$
$$J_p = \{j \in [n] \mid F_i \cap G_j \neq \emptyset, z_j < 1\}$$

Suppose that $|J_f| = m$ and suppose that J_p is sorted as $J_p = \{j_1, \dots, j_t\}$ where $j_1 \leq j_2 \leq \dots \leq j_t$. For each $l = 1, \dots, t+1$ define

$$u_l = y(F_i \cap G_{j_l}) + y(F_i \cap G_{j_{l+1}}) + \dots + y(F_i \cap G_{j_t})$$

Then we have $1 \geq u_1 \geq u_2 \geq \dots \geq u_t \geq u_{t+1} = 0$, $m \geq 1$, and

$$\mathbf{E}[T_i \mid Q_p, Q_f] \leq R \Bigg(1 + \Big(1 - \frac{(1 - Q_f)(1 - u_1)}{m} \Big)^m \prod_{l=1}^{t} (1 - (1 - Q_p)(u_l - u_{l+1})) + \Big(1 - \frac{(1 - Q_f)(1 - u_1)}{m} - Q_f \Big)^m \prod_{l=1}^{t} (1 - u_l + (1 - Q_p) u_l) \Bigg).$$

Proof. In this proof, we will condition on a fixed value for $Q_{\mathrm{f}}, Q_{\mathrm{p}}$. All probabilities should be interpreted as conditioned on these values; we will not note this explicitly for the remainder of the proof.

For $l = 1, \dots, t$ we let $r_l = y(F_i \cap G_{j_l}) = u_l - u_{l+1}$. For $j \in J_{\mathrm{f}}$, we let $s_j = y(F_i \cap G_j)$. Observe that as $\sum_j y(F_i \cap G_j) = 1$, we have $\sum_j s_j + \sum_l r_l = 1$.

A necessary condition for $T_i \geq 2R$ is that we do not open any center in F_i. Applying Proposition C.2 with $S = F_i$ yields

$$\begin{aligned} P(\mathcal{S} \cap F_i = \emptyset) &\leq \prod_{l=1}^{t} (1 - (1 - Q_{\mathrm{p}}) y(F_i \cap G_{j_l}) - Q_{\mathrm{p}} z_{j_l} [c_{j_l} \in F_i]) \prod_{j \in J_{\mathrm{f}}} (1 - (1 - Q_{\mathrm{f}}) y(F_i \cap F_j) - Q_{\mathrm{f}} [j \in F_i]) \\ &\leq \prod_{l=1}^{t} (1 - (1 - Q_{\mathrm{p}}) y(F_i \cap G_{j_l})) \prod_{j \in J_{\mathrm{f}}} (1 - (1 - Q_{\mathrm{f}}) y(F_i \cap F_j)) \\ &= \prod_{l=1}^{t} (1 - (1 - Q_{\mathrm{p}}) r_l) \prod_{j \in J_{\mathrm{f}}} (1 - (1 - Q_{\mathrm{f}}) s_j) \end{aligned}$$

A necessary condition for $T_i \geq 3R$ is that we do not open any center in F_i, nor do we open the cluster center of any cluster which intersects with F_i. Applying Proposition C.2 with $S = F_i \cup \{c_j \mid G_j \cap F_i \neq \emptyset\}$ yields:

$$\begin{aligned} P(\mathcal{S} \cap S = \emptyset) &\leq \prod_{l=1}^{t} (1 - (1 - Q_{\mathrm{p}}) y(S \cap G_{j_l}) - Q_{\mathrm{p}} z_{j_l} [c_{j_l} \in S]) \prod_{j \in J_{\mathrm{f}}} (1 - (1 - Q_{\mathrm{f}}) y(S \cap F_j) - Q_{\mathrm{f}} [j \in S]) \\ &\leq \prod_{l=1}^{t} (1 - (1 - Q_{\mathrm{p}}) y(F_i \cap G_{j_l}) - Q_{\mathrm{p}} z_l) \prod_{j \in J_{\mathrm{f}}} (1 - (1 - Q_{\mathrm{f}}) y(F_i \cap F_j) - Q_{\mathrm{f}}) \\ &= \prod_{l=1}^{t} (1 - (1 - Q_{\mathrm{p}}) r_l - Q_{\mathrm{p}} z_{j_l}) \prod_{j \in J_{\mathrm{f}}} (1 - (1 - Q_{\mathrm{f}}) s_j - Q_{\mathrm{f}}) \end{aligned}$$

Thus, we have that

$$\begin{aligned} \mathbf{E}[T_i] \leq R \Bigg(1 &+ \prod_{l=1}^{t} (1 - (1 - Q_{\mathrm{p}}) r_l) \prod_{j \in J_{\mathrm{f}}} (1 - (1 - Q_{\mathrm{f}}) s_j) \\ &+ \prod_{l=1}^{t} (1 - (1 - Q_{\mathrm{p}}) r_l - Q_{\mathrm{p}} z_{j_l}) \prod_{j \in J_{\mathrm{f}}} (1 - (1 - Q_{\mathrm{f}}) s_j - Q_{\mathrm{f}}) \Bigg). \end{aligned} \tag{26}$$

By the AM-GM inequality, we have

$$\prod_{j \in J_{\mathrm{f}}} (1 - (1 - Q_{\mathrm{f}}) s_j) \leq \left(1 - (1 - Q_{\mathrm{f}}) \frac{\sum_{j \in J_{\mathrm{f}}} s_j}{m} \right)^m = \left(1 - (1 - Q_{\mathrm{f}}) \frac{1 - u_1}{m} \right)^m,$$

and similarly

$$\prod_{j \in J_{\mathrm{f}}} (1 - (1 - Q_{\mathrm{f}}) s_j - Q_{\mathrm{f}}) \leq \left(1 - (1 - Q_{\mathrm{f}}) \frac{1 - u_1}{m} - Q_{\mathrm{f}} \right)^m.$$

This gives

$$\mathbf{E}[T_i] \leq R\bigg(1 + \Big(1 - (1 - Q_{\mathrm{f}})\frac{1 - u_1}{m}\Big)^m \prod_{l=1}^{t}(1 - (1 - Q_{\mathrm{p}})r_l) + \Big(1 - (1 - Q_{\mathrm{f}})\frac{1 - u_1}{m} - Q_{\mathrm{f}}\Big)^m \prod_{l=1}^{t}(1 - (1 - Q_{\mathrm{p}})r_l - Q_{\mathrm{p}} z_{j_l})\bigg). \tag{27}$$

We next claim that $z_{j_l} \geq u_l$ for all $l = 1, \dots, t$. For, by Proposition C.1, we have

$$\begin{aligned} z_{j_l} &\geq y(F_i - F_{\pi(1)} - \dots - F_{\pi(j_l - 1)}) \\ &\geq y(F_i) - \sum_{j \in J_{\mathrm{f}}} y(F_i \cap G_j) - \sum_{v < j_l} y(F_i \cap G_v) \\ &= y(F_i) - \sum_{j \in J_{\mathrm{f}}} y(F_i \cap G_j) \sum_{v < l} y(F_i \cap G_{j_v}) \qquad \text{as } F_i \cap G_v = \emptyset \text{ for } v \notin J_{\mathrm{p}} \cup J_{\mathrm{f}} \\ &= 1 - \sum_{j \in J_{\mathrm{f}}} s_j - r_1 - \dots - r_{l-1} = u_l. \end{aligned}$$

Observe that if $m = 0$, then $u_1 = 1$; but this implies that $z_{j_1} \geq u_1 = 1$, which contradicts $z_{j_1} < 1$. Thus, we have shown that $m \geq 1$ as desired. Finally, the claim follows as (27) is a decreasing function of each parameter z_{j_l} and $r_l = u_{l+1} - u_l$.

□

We will use Proposition C.3 to bound $\mathbf{E}[T_i]$, over all possible integer values $m \geq 1$ and over all possible sequences $u_1 \geq u_2 \geq u_3 \geq \dots \geq u_t \geq 0$. One difficulty with this is that this is not a compact space; it fails for two reasons: because the dimension t is unbounded, and because m can have a potentially infinite set of values. The next results removes these restrictions.

Proposition C.4. *For any fixed integers $L, M \geq 1$, we have the bound*

$$\mathbf{E}[T_i] \leq R\Big(1 + \max_{\substack{m \in \{1,2,\dots,M\} \\ 1 \geq u_1 \geq a_2 \geq \dots u_L \geq 0}} \mathbf{E}_Q \hat{T}_{L,M}(m, u_1, u_2, \dots, u_L)\Big),$$

where we define

$$\alpha = \prod_{l=1}^{L-1}(1 - (1 - Q_p)(u_l - u_{l+1})) \times e^{-(1-Q_p)u_L}$$

$$\beta = \prod_{l=1}^{L-1}(1 - u_l + (1 - Q_p)u_{l+1}) \times \begin{cases} (1 - u_L) & \text{if } u_L \leq Q_p \\ e^{-\frac{u_L - Q_p}{1 - Q_p}}(1 - Q_p) & \text{if } u_L > Q_p \end{cases}$$

$$\hat{T}_{L,M}(m, u_1, \dots, u_L) = \begin{cases} (1 - (1 - Q_f)(1 - u_1)/m)^m \alpha + (1 - (1 - Q_f)(1 - u_1)/m - Q_f)^m \beta & \text{if } m < M \\ e^{-(1-Q_f)(1-u_1)}\alpha + (1 - Q_f)^M e^{-(1-u_1)}\beta & \text{if } m = M \end{cases}$$

The expectation $\mathbf{E}_Q$ is taken only over the randomness involved in Q_f, Q_p.

Proof. By Proposition C.3, we have that

$$\mathbf{E}[T_i \mid Q_{\mathrm{p}}, Q_{\mathrm{f}}] \leq R\Bigg(1 + \Big(1 - \frac{(1 - Q_{\mathrm{f}})(1 - u_1)}{m}\Big)^m \prod_{l=1}^{t}(1 - (1 - Q_{\mathrm{p}})(u_l - u_{l+1})) + \Big(1 - \frac{(1 - Q_{\mathrm{f}})(1 - u_1)}{m} - Q_{\mathrm{f}}\Big)^m \prod_{l=1}^{t}(1 - u_l + (1 - Q_{\mathrm{p}})u_l)\Bigg).$$

where $u_1, \ldots, u_t, m$ are defined as Proposition C.3; in particular $1 \geq u_1 \geq u_2 \geq \cdots \geq u_t \geq u_{t+1} = 0$ and $m \geq 1$.

Now let us define $u_j = 0$ for all integers $j \geq t$. Then we have that

$$\begin{aligned}\mathbf{E}[T_i \mid Q_{\mathrm{p}}, Q_{\mathrm{f}}] \leq R\Bigg(1 + \left(1 - \frac{(1-Q_{\mathrm{f}})(1-u_1)}{m}\right)^m \prod_{l=1}^{\infty}(1-(1-Q_{\mathrm{p}})(u_l - u_{l+1})) \\ + \left(1 - \frac{(1-Q_{\mathrm{f}})(1-u_1)}{m} - Q_{\mathrm{f}}\right)^m \prod_{l=1}^{\infty}(1 - u_l + (1-Q_{\mathrm{p}})u_l)\Bigg).\end{aligned}$$

We upper-bound the terms corresponding to $l > L$ as follows:

$$\begin{aligned}\prod_{l=L}^{\infty}(1-(1-Q_{\mathrm{p}})(u_l - u_{l+1})) &\leq \prod_{l=L}^{\infty}\exp(-(1-Q_{\mathrm{p}})(u_l - u_{l+1})) \\ &= \exp(-(1-Q_{\mathrm{p}})\sum_{l=L}^{\infty}(u_l - u_{l+1})) \\ &= \exp(-(1-Q_{\mathrm{p}})u_L) \qquad \text{(sum telescopes)}\end{aligned}$$

and

$$\begin{aligned}\prod_{l=L}^{\infty}(1 - u_l + (1-Q_{\mathrm{p}})u_{l+1}) &\leq (1 - u_L + (1-Q_{\mathrm{p}})u_{L+1})\prod_{l=L+1}^{\infty}\exp(-u_l + (1-Q_{\mathrm{p}})u_{l+1}) \\ &= (1 - u_L + (1-Q_{\mathrm{p}})u_{L+1})e^{-u_{L+1}}e^{\sum_{l=L+2}^{\infty}(-Q_{\mathrm{p}}u_l)} \\ &\leq (1 - u_L + (1-Q_{\mathrm{p}})u_{L+1})e^{-u_{L+1}}.\end{aligned}$$

Now consider the expression $(1 - u_L + (1-Q_{\mathrm{p}})u_{L+1})e^{-u_{L+1}}$ as a function of u_{L+1}. Elementary calculus shows that for $u_{L+1} \in [0, u_L]$ it achieves a maximum value $u_{L+1} = \max(0, \frac{u_L - Q_{\mathrm{p}}}{1-Q_{\mathrm{p}}})$. Substituting in this value gives us that, when $u_L \leq Q_{\mathrm{p}}$ we have

$$\prod_{l=L}^{\infty}(1 - u_l + (1-Q_{\mathrm{p}})u_{l+1}) \leq (1 - u_L),$$

and similarly when $u_L > Q_{\mathrm{p}}$ we have

$$\prod_{l=L}^{\infty}(1 - u_l + (1-Q_{\mathrm{p}})u_{l+1}) \leq e^{-\frac{u_L - Q_{\mathrm{p}}}{1-Q_{\mathrm{p}}}}(1-Q_{\mathrm{p}}).$$

Thus we have

$$\mathbf{E}[T_i \mid Q_{\mathrm{f}}, Q_{\mathrm{p}}] \leq R\left(1 + \left(1 - \frac{(1-Q_{\mathrm{f}})(1-u_1)}{m}\right)^m \alpha + \left(1 - \frac{(1-Q_{\mathrm{f}})(1-u_1)}{m} - Q_{\mathrm{f}}\right)^m \beta\right).$$

If $m < M$ we are done. Otherwise, for $m \geq M$, we upper bound the Q_{f} terms as:

$$\begin{aligned}(1-(1-Q_{\mathrm{f}})(1-u_1)/m)^m &\leq \exp(-(1-Q_{\mathrm{f}})(1-u_1)/m)^m \\ &= e^{-(1-Q_{\mathrm{f}})(1-u_1)},\end{aligned}$$

and

$$\begin{aligned}(1-(1-Q_{\mathrm{f}})(1-u_1)/m-Q_{\mathrm{f}})^m &= (1-Q_{\mathrm{f}})^m(1-(1-u_1)/m)^m \\ &\leq (1-Q_{\mathrm{f}})^M \exp(-(1-u_1)/m)^m \\ &= (1-Q_{\mathrm{f}})^M \exp(-(1-u_1)),\end{aligned}$$

and we thus have that $\mathbf{E}[T_i \mid Q_{\mathrm{f}}, Q_{\mathrm{p}}] \leq R(1+\hat{T}_{L,M}(M,u_1,\dots,u_L))$.

□

We now discuss to bound $\hat{T}_{L,M}$, where we select $Q_{\mathrm{f}}, Q_{\mathrm{p}}$ according to the following type of distribution:

$$(Q_{\mathrm{f}}, Q_{\mathrm{p}}) = \begin{cases}(\gamma_{0,\mathrm{f}}, 0) & \text{with probability } p \\ (\gamma_{1,\mathrm{f}}, \gamma_{1,\mathrm{p}}) & \text{with probability } 1-p\end{cases}$$

For this distribution, it is straightforward to calculate $\mathbf{E}_Q \hat{T}_{L,M}(m,u_1,\dots,u_L)$ for any fixed $u_1,\dots,u_L,m$. Now suppose we want to upper-bound it over the compact domain $m \in \{1,\dots,M\}, 1 \geq u_1 \geq \cdots \geq u_L \geq 0$. The most straightforward way to show this would be to divide $u_1,\dots,u_L$ into intervals of size ϵ. We then enumerate over all possible m and possible intervals for $u_1,\dots,u_L$ and use interval arithmetic to calculate an upper bound on $\hat{T}_{L,M}$. However, this would have a running time ϵ^{-L} which is excessive.

But we make the following crucial observation: suppose we have fixed some $u_j,\dots,u_L$, and we wish to continue to enumerate over $u_1,\dots,u_L$. To compute $\hat{T}(m,u_1,\dots,u_L)$ as a function of $m,u_1,\dots,u_L$ we do not need to know the precise value of $u_{j+1},\dots,u_L$, but only the following four quantities:

1. $e^{-(1-\gamma_{1,\mathrm{p}})u_L}\prod_{l=j}^{L-1}(1-(1-\gamma_{1,\mathrm{p}})(u_l-u_{l+1}))$,
2. $\prod_{l=j}^{L-1}(1-u_l+(1-\gamma_{1,\mathrm{p}})u_{l+1}) \times \begin{cases}(1-u_L) & \text{if } u_L \leq \gamma_{1,\mathrm{p}} \\ e^{-\frac{u_L-\gamma_{1,\mathrm{p}}}{1-\gamma_{1,\mathrm{p}}}}(1-\gamma_{1,\mathrm{p}}) & \text{if } u_L > \gamma_{1,\mathrm{p}}\end{cases}$,
3. $e^{-u_L}\prod_{l=j}^{L-1}(1-(u_l-u_{l+1}))$,
4. u_{j+1}.

Thus, we can use a dynamic programming approach: for $j = L,\dots,1$, we compute all possible values for these terms in a recursive fashion.[1] Furthermore, we only need to keep track of the *maximal* four-tuples for these four quantities. The resulting search space has size $O(\epsilon^{-3})$.

Theorem 4.3. *Suppose we set our parameters for the distribution on Q_f, Q_p as follows:*

$$(Q_f, Q_p) = \begin{cases}(0.4525, 0) & \textit{with probability } p = 0.773436 \\ (0.0480, 0.3950) & \textit{with probability } 1-p\end{cases}.$$

Then, $T_i \leq 3R$ with probability one, and $\mathbf{E}[T_i] \leq 1.592R$.

Proof. We maximize $\hat{T}_{L,M}(m,u_1,\dots,u_L)$ with $M = 10, \epsilon = 2^{-12}, L = 7$. We wrote a C code for this computation; it runs in about an hour on a single CPU core. With some optimizations, it is possible to optimize over the parameter $p \in [0,1]$ while still keeping the stack space bounded by $O(\epsilon^{-3})$.

[1] We note that we restrict $\gamma_{0,\mathrm{p}} = 0$ in order to keep this search space controlled. If $\gamma_{\mathrm{p}} > 0$, then we would need to track an additional term as well, making the search space infeasibly large.

(We note that our code uses interval arithmetic to calculate an upper bound on $\hat{T}_{L,M}$, but the calculations are carried out using double-precision floating point arithmetic. This is not completely rigorous because we are using the default floating-point rounding instead of the special rounding modes. While the errors committed by this rounding were not tracked explicitly, we believe that they should be orders of magnitude below the third decimal place, and so should not affect the validity of our results.)

$\square$